



A sequential Benders-based mixed-integer quadratic programming algorithm and its implementation in the CAMINO toolbox

Andrea Ghezzi¹ · Wim Van Roy² · Sebastian Sager^{3,4} · Moritz Diehl⁵

Received: 20 August 2025 / Accepted: 2 June 2026
© The Author(s) 2026, modified publication 2026

Abstract

Sequential quadratic programming and sequential convex programming efficiently solve nonlinear programs (NLPs) by linearizing inner nonlinearities while preserving the outer convex structure. This paper introduces a sequential mixed-integer quadratic programming (MIQP) algorithm to extend this methodology to mixed-integer nonlinear problems (MINLPs), leveraging the efficiency of modern MIQP solvers. The algorithm uses a three-step iterative process. First, the MINLP is linearized around the current iterate. Second, an MIQP is formulated and solved, with its feasible region restricted to a specific area around the linearization point. This region is defined using objective values and derivatives from previous iterations, drawing on concepts from generalized Benders' decomposition. Third, the integer variables from the MIQP solution are fixed, and an NLP involving only the continuous variables is solved. The best solution among all iterates becomes the linearization point for the next iteration. A fallback strategy based on a mixed-integer linear program (MILP) is used when MIQP progress stalls. This guarantees convergence to the globally optimal solution

A. Ghezzi and W. Van Roy contributed equally.

✉ Andrea Ghezzi
andrea.ghezzi@imtek.uni-freiburg.de

✉ Wim Van Roy
wim.vanroy@kuleuven.be

Sebastian Sager
sager@ovgu.de

Moritz Diehl
moritz.diehl@imtek.uni-freiburg.de

¹ Department of Microsystems Engineering (IMTEK), University of Freiburg, Freiburg, Germany

² Department of Mechanical Engineering, KU Leuven, Leuven, Belgium

³ Institute of Mathematical Optimization, Otto-von-Guericke Universität Magdeburg, Magdeburg, Germany

⁴ Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

⁵ Department of Microsystems Engineering (IMTEK) and Department of Mathematics, University of Freiburg, Freiburg, Germany

for convex MINLPs. For nonconvex problems, the algorithm functions as a heuristic without global optimality guarantees. Numerical experiments show its competitiveness with other MINLP solvers on benchmark problems. In addition, the algorithm was successfully applied to mixed-integer optimal control problems, demonstrating its effectiveness in handling challenging nonlinear equality constraints. The proposed algorithm is publicly available at <https://github.com/minlp-toolbox/CAMINO> with the name `s-b-miqp`.

Keywords Mixed-integer nonlinear programming (MINLP) · Sequential mixed-integer quadratic programming · Optimal control · Open-source scientific software

Mathematics Subject Classification 90C11 · 90C30 · 90C59 · 65K05 · 49M25 · 49-04

1 Introduction

The class of mixed-integer nonlinear programs (MINLPs) comprises problems characterized by continuous and discrete variables coupled with nonlinear relationships in the objective or constraints. Hence, MINLPs represent a powerful optimization paradigm that offers a natural way to formulate a wide range of problems and applications. The intersection of nonlinearity and discrete variables poses unique challenges for the numerical solution of such problems, which are characterized by NP-hard complexity [30, 39]. For unbounded support, they are even undecidable, meaning that not every unbounded MINLP problem can be solved.

Several excellent surveys and books cover algorithms for solving MINLPs [7], [57, §21]. Here, we provide only a high-level picture of the field and focus on the literature directly connected to the algorithm we propose.

One can divide the existing algorithms into two families. The first consists of branch-and-bound-type algorithms, such as nonlinear branch-and-bound [22, 33] and spatial branch-and-bound [58]. The second family relies on the creation of cutting planes to iteratively tighten the integer search space, as in the generalized Benders' decomposition (GBD) [31], outer approximation [25] and its quadratic version [28], and extended supporting cutting planes [40, 62]. A combination of both families includes LP/NLP-based branch-and-bound [2, 52] and branch-and-check [59]. Branch-and-bound-based methods are appealing because they leverage the existence of reliable, efficient, and open-source nonlinear solvers such as IPOPT [61] and WORHP [18]. In contrast, the second family of methods additionally requires efficient mixed-integer linear (quadratic) solvers such as the open-source CBC [29], SCIP [3], HiGHS [37], as well as the commercial CPLEX [38], Gurobi [34], and MOSEK [51]. Moreover, there also exist solvers that directly implement MINLP-specific algorithms, such as the open-source Bonmin [11], Couenne [6], SHOT [46], and the commercial Antigone [50], Baron [56], Knitro [21], and Gurobi. A sequential programming algorithm for mixed-integer programs was suggested in [26], resembling a standard trust-region sequential quadratic programming (SQP) method for continuous decision spaces. The algorithm in [26] can be applied to *nonconvex* MINLPs and also to MINLPs where the integer variables cannot be relaxed. However, the authors do not provide a proof

of convergence to the global minimizer, even for *convex* MINLPs. The Mittelman benchmarks webpage [47] gives an up-to-date overview of the current solvers and their performance.

Most generic MINLP solvers focus on solving *convex* MINLPs, often providing only heuristics for *nonconvex* ones. In fact, the solution of *nonconvex* MINLPs is much harder, as it requires the construction of global underestimators. Most solvers implement a spatial branch-and-bound algorithm, which can be built on top of efficient branch-and-bound codes. For decomposition approaches, a method to extend GBD for solving *nonconvex* MINLPs is presented in [44]. This method decomposes the original *nonconvex* MINLP into convex subproblems by generating tight convex relaxations. In general, adopting standard decomposition methods for solving *nonconvex* MINLPs may fail, particularly in the presence of numerous equality constraints. A practical example where these constraints arise is in the solution of mixed-integer optimal control problems (MIOCPs) via direct methods, such as direct collocation [60] or direct multiple shooting [10]. In such cases, equality constraints enforce continuity of the underlying dynamics equations at every grid node.

For the solution of MIOCPs, an important class of methods known as combinatorial integral approximation (CIA) [55] employs an error-controlled decomposition approach. In practice, the MIOCP is discretized to obtain an MINLP, which is then approximately solved by computing an integer approximation. This approximation minimizes the distance from the relaxed solution of the MINLP using a dedicated norm. An open-source implementation of this approach, named pycombina [15], has been developed and demonstrated to be effective in various engineering applications [16, 54]. This method is capable of handling generic dwell-time constraints [63] and provides fast approximate solutions, with the main computational challenge lying in the nonlinear optimization rather than combinatorial aspects. The method relies on the similarity of the relaxed solution with the optimal one. This is a drawback when dealing with coarse discretization grids or long uptimes, as shown in [1, 17]. To address this, [17] proposes a new distance function for the second step, based on a quadratic programming approximation around the relaxed MINLP solution. This approach, which is related to a single iteration of the method presented in this paper, often enhances the quality of the MIOCP solution in terms of both objective and constraint satisfaction.

Another class of methods for approximately solving MIOCP is switching time optimization (STO). This approach relies on an initial sequence of states that can be applied to the system, with switching times optimized. During the iterations, the sequence is adjusted either by inserting additional elements [5, 42] or by removing unnecessary sequences [1]. Also, this approach is a heuristic and relies on the initial sequence provided to the solver.

1.1 Contribution

In this work, we introduce an algorithm for solving MINLPs, drawing inspiration from SQP and classical MINLP methods based on outer approximation. We obtain a mixed-integer quadratic program (MIQP) master problem by linearizing the original MINLP at the best solution found. The integer part of the MIQP solution is then fixed

in the MINLP to obtain a nonlinear program (NLP). After each MIQP-NLP iteration, we compute a new GBD cut that is used to construct a polytope, named “Benders region”. The Benders region further restricts the feasible set of the MIQP to a portion where we expect to obtain an accurate MIQP approximation of the given MINLP. To guarantee convergence to global minima in case of *convex* MINLPs, we introduce a second master problem that takes the form of a MILP, which contains OA cuts for the best solution found and GBD cuts for all the other visited points. This MILP has the property of underestimating the original MINLP, thus providing a valid lower bound for the MINLP objective. For *nonconvex* MINLPs, the presented algorithm is a heuristic.

Unlike quadratic OA [28], our MIQP master problem contains a linearization of the constraints of the original MINLP only for the best solution found. The other constraints correspond to the Benders region, based on GBD cuts, and infeasibility (or *no-good*) cuts. Also, the additional MILP master problem allows us to prove convergence to the global minimizer for convex MINLPs. We generate infeasibility cuts by projecting the infeasible integer point on the relaxed feasible set rather than minimizing a norm of the constraint violation as done in [25] or [28]. In our preliminary work [32], we presented the idea of sequentially solving mixed-integer programs (MIPs) while restricting their feasible set using polytopical regions computed with the information collected during algorithm iterations. However, in [32] the proposed algorithm is a pure heuristic even for *convex* MINLPs. The polytopical regions do not leverage gradient information as they are based on a Voronoi partitioning, and we did not include a mechanism to handle infeasible NLPs as we assumed that they were always feasible by using slack variables.

A fundamental contribution of this work is the open-source implementation of the proposed algorithm within the package CAMINO¹ (Collection of Algorithms for Mixed-Integer Nonlinear Optimization). CAMINO is written in Python and relies on CasADi [4] to both model optimization problems and interface existing solvers. The users have the flexibility to choose from various solvers interfaced by CasADi, to solve both NLPs and MIPs, enabling an implementation free from reliance on commercial solvers. In addition to the new algorithm presented in this paper, CAMINO includes a comprehensive range of other existing algorithms, such as GBD, (quadratic) outer approximation, feasibility pumps, and the ability to seamlessly use MINLP solvers, already interfaced by CasADi, such as Bonmin. Further details are available in the README file of the repository.

1.2 Outline

In the remainder of this section, we outline some preliminary definitions and notions utilized throughout the rest of the paper. In Sect. 2, we introduce the new algorithm, first by giving a short overview, and later describing in detail all its constituent components. Moreover, we prove that for *convex* MINLPs the algorithm converges in a finite number of iterations either to the global optimum or to a certificate of infeasibility. We conclude the section by illustrating the behavior of the proposed algorithm with

¹ <https://github.com/minlp-toolbox/CAMINO>.

a simple tutorial example. In Sect. 3, we propose an extension for treating nonconvex MINLPs by introducing heuristics to modify the generated cutting planes. We demonstrate that these introduced heuristics do not compromise convergence to the global minimizer in the case of *convex* MINLPs. Additionally, for MINLPs where the integer variables enter affinely, making the relaxed integer feasible set convex, we prove that the proposed algorithm terminates either at a feasible solution or with a certificate of infeasibility. In Sect. 4, we compare the proposed algorithm against Bonmin, Gurobi, SCIP, and SHOT on a large subset of MINLPs instances from the MINLPLib [19, 49] containing at least one integer and one continuous variable. Moreover, we present the results obtained with the proposed algorithm in two cases of optimal control for switched systems: a textbook example of a small, nonlinear, and unstable system, and a complex nonlinear energy system for building control. For the latter example, we could only compare against Bonmin and the specialized algorithm CIA. Finally, Sect. 5 presents some conclusions and future work directions.

1.3 Notation

We denote with $\mathbb{Z}_{[a,b]}$ the set of integer numbers in the interval $[a, b]$ with $a, b \in \mathbb{Z}$ and $a < b$, and with $\mathbb{Z}_{\geq 0}$, the set of nonnegative integer numbers. For a vector-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, we denote the transpose of the Jacobian as $\nabla f(z) = \left(\frac{\partial f}{\partial z}(z) \right)^\top$, such that $\nabla f(z) \in \mathbb{R}^{n \times m}$. The term *convex* MINLP refers to a MINLP that is convex with respect to both the continuous optimization variables and the relaxed integer variables.

1.4 Mixed-integer nonlinear problem formulations

We consider the generic MINLP with $x \in X \subset \mathbb{R}^n$ and $y \in Y \subset \mathbb{Z}^m$ of the form

$$\begin{aligned} \mathcal{P}_{\text{MINLP}} : \quad & \min_{x \in X, y \in Y} f(x, y) \\ & \text{s.t.} \quad g(x, y) \leq 0, \\ & \quad \quad h(x, y) = 0. \end{aligned} \quad (1)$$

Minimizing over the continuous variable $x \in X$ for a fixed $y \in Y$ yields the following parametric NLP

$$\begin{aligned} \mathcal{P}_{\text{NLP}} : \quad & J(y) := \min_{x \in X} f(x, y) \\ & \text{s.t.} \quad g(x, y) \leq 0, \\ & \quad \quad h(x, y) = 0, \end{aligned} \quad (2)$$

Now, we can conceptually state $\mathcal{P}_{\text{MINLP}}$ as

$$\min_{y \in Y} J(y). \quad (3)$$

We highlight the definition of J since the proposed algorithm solves $\mathcal{P}_{\text{MINLP}}$ exploiting function evaluations and first-order information of J itself. By slight abuse of notation, we use $J(y)$ to denote either the optimization problem $\mathcal{P}_{\text{NLP}}$, its objective function, or a specific objective value. The intended meaning should be clear from the context.

The proofs in this work rely on the following assumptions regarding problem (1).

Assumption 1 We assume that

1. $Y = \bar{Y} \cap \mathbb{Z}^{n_y}$, where both $X \subset \mathbb{R}^{n_x}$ and $\bar{Y} \subset \mathbb{R}^{n_y}$ are closed convex polyhedral sets.
2. Functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_g}$ and $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_h}$ are at least twice continuously differentiable.
3. The integer set Y is finite.

Assumption 2 The Mangasarian-Fromovitz constraint qualification holds at the solution x^* of $\mathcal{P}_{\text{NLP}}$ with $J(y)$ for all $y \in Y$.

Definition 1 We define the feasible set of the integer variables as

$$\mathcal{F} := \{y \in Y \mid \exists x \in X, g(x, y) \leq 0, h(x, y) = 0\}.$$

Similarly, we define the feasible set of the relaxed integer variables as

$$\bar{\mathcal{F}} := \{y \in \bar{Y} \mid \exists x \in X, g(x, y) \leq 0, h(x, y) = 0\}.$$

Definition 2 Let f be a continuously differentiable function. We define its linearization at $(\bar{x}, \bar{y})$ as a first-order Taylor expansion $f_L(x, y; \bar{x}, \bar{y}) := f(\bar{x}, \bar{y}) + \nabla f(\bar{x}, \bar{y})^\top \begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix}$.

Assumption 3 (Convexity) Function h is affine, and functions $g_1, \dots, g_{n_g}$ and f are convex on $X \times \bar{Y}$.

With Assumption 3, the set $\bar{\mathcal{F}}$ would be convex and any subgradient of J would provide a lower bound of J .

2 Sequential benders MIQP algorithm

The algorithm proposed in this work operates by leveraging successive approximations of the original MINLP problem (1), which are constructed using first- and second-order derivative information of the problem functions. At each iteration, a quadratic master problem, denoted as $\mathcal{P}_{\text{BR-MIQP}}$, is formulated by linearizing $\mathcal{P}_{\text{MINLP}}$ around the best

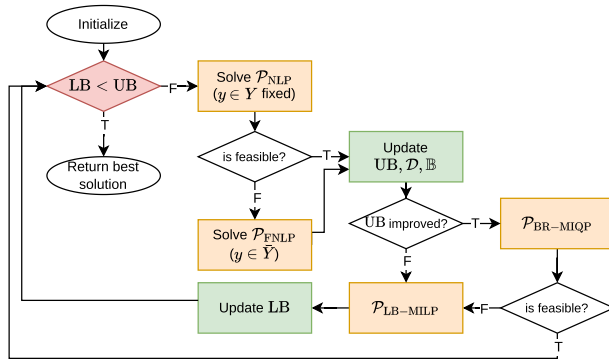


Fig. 1 High-level flow chart of the proposed algorithm. The orange rectangles highlight the optimization problems solved during the algorithm iterations, the green rectangles perform operations on the data obtained by solving the different problems, and the red diamond highlights the termination condition. The outcome “T” of the diamonds corresponds to “True”, and “F” corresponds to “False”. For a rigorous description, see the pseudocode of Algorithm 1

solution visited so far, (x_b, y_b) . This approximation is expected to provide high-quality candidate solutions within a defined region $\mathbb{B}$ around the current linearization point. In parallel, a second master problem, $\mathcal{P}_{\text{LB-MILP}}$, is constructed using accumulated first-order information from all previously visited points, following a similar philosophy to generalized Benders decomposition [31]. These two master problems jointly address the combinatorial aspects of $\mathcal{P}_{\text{MINLP}}$ while providing complementary perspectives for exploring the feasible space. To evaluate the quality of the candidate solutions with respect to the original problem, an auxiliary nonlinear programming problem $\mathcal{P}_{\text{NLP}}$ is solved. We include a second auxiliary problem $\mathcal{P}_{\text{FNLP}}$, which is solved whenever $\mathcal{P}_{\text{NLP}}$ is infeasible. The overall procedure is illustrated in Fig. 1, where a single iteration is represented by a full cycle from the red diamond labeled “LB < UB” back to itself. The iteration index is denoted by $k \in \mathbb{Z}_{\geq 0}$. As evident from the flowchart, the algorithm follows the general structure of outer approximation methods, but distinguishes itself by employing two separate master problems.

Moving forward, we first present the constituent components of the algorithm highlighted in Fig. 1, namely the auxiliary problems, the master problems, and the termination condition. Secondly, by means of pseudocode, we present the algorithm in detail and derive its theoretical properties. We close the section with a tutorial example illustrating the algorithm’s behavior.

2.1 Computation of $\nabla J(y)$ from the solution of the auxiliary problem $\mathcal{P}_{\text{NLP}}$

The auxiliary problem $\mathcal{P}_{\text{NLP}}$ (2), introduced in Sect. 1.4, is obtained by fixing the integer variables of the original MINLP (1). For a given $\tilde{y} \in Y$, solving $\mathcal{P}_{\text{NLP}}$ yields the optimal value $J(\tilde{y})$. If $\mathcal{P}_{\text{NLP}}$ is infeasible, we set $J(\tilde{y}) = +\infty$. The function J is generally nonsmooth due to changes in the active set and possible nonuniqueness of optimal solutions of $\mathcal{P}_{\text{NLP}}$. Consequently, classical differentiability cannot be expected, and generalized derivatives must be employed.

Under Assumption 2 (MFCQ), the set of Lagrange multipliers associated with a local solution is nonempty and bounded, and the value function J is locally Lipschitz continuous around $\tilde{y}$. To characterize the first-order sensitivity of J with respect to y , we consider a lifted formulation of $\mathcal{P}_{\text{NLP}}$ in which y is treated as an optimization variable and fixed by an equality constraint

$$\begin{aligned} \min_{x \in X, y \in Y} \quad & f(x, y) \\ \text{s.t.} \quad & g(x, y) \leq 0, \\ & h(x, y) = 0, \\ & y - \tilde{y} = 0. \end{aligned} \quad (4)$$

Let $(x^*, \tilde{y})$ be a locally optimal solution of (4), and let $\lambda \in \mathbb{R}^{n_g}$, $\mu \in \mathbb{R}^{n_h}$, and $\mu_{\tilde{y}} \in \mathbb{R}^{n_y}$ denote the Lagrange multipliers associated with the inequality constraints, equality constraints, and the constraint $y - \tilde{y} = 0$, respectively. The corresponding Karush-Kuhn-Tucker (KKT) conditions are

$$\begin{cases} \nabla_x f(x^*, \tilde{y}) + \nabla_x g(x^*, \tilde{y})\lambda + \nabla_x h(x^*, \tilde{y})\mu &= 0, \\ \nabla_y f(x^*, \tilde{y}) + \nabla_y g(x^*, \tilde{y})\lambda + \nabla_y h(x^*, \tilde{y})\mu &= -\mu_{\tilde{y}}, \\ g(x^*, \tilde{y}) &\leq 0, \\ h(x^*, \tilde{y}) &= 0, \\ \lambda &\geq 0, \\ \lambda_i g_i(x^*, \tilde{y}) &= 0, \quad i = 1, \dots, n_g. \end{cases} \quad (5)$$

Proposition 1 (Clarke subgradient of the value function under MFCQ) *Suppose that MFCQ holds at $(x^*, \tilde{y})$ for problem (4), and that $(x^*, \tilde{y})$ is a locally optimal solution. Then the value function J is locally Lipschitz continuous around $\tilde{y}$ and*

$$-\mu_{\tilde{y}} \in \partial_C J(\tilde{y}),$$

where $\partial_C J(\tilde{y})$ denotes the Clarke subdifferential of J at $\tilde{y}$.

Proof Under MFCQ, the set of Lagrange multipliers of (4) is nonempty and bounded. Moreover, classical sensitivity results for parametric nonlinear programs imply that the value function is locally Lipschitz and that the multiplier associated with the parameter-fixing constraint characterizes the first-order variation of the optimal value. See, e.g., [27, Ch. 3] for a detailed proof. $\square$

As a consequence of Proposition 1, we define

$$\nabla J(\tilde{y}) := -\mu_{\tilde{y}}, \quad (6)$$

where $\nabla J(\tilde{y})$ is interpreted as an element of the Clarke subdifferential $\partial_C J(\tilde{y})$.

Remark 1 Because J is nonsmooth, $\nabla J(\tilde{y})$ is not a gradient in the classical sense. When $\mathcal{P}_{\text{NLP}}$ is solved using an interior-point method, the computed multipliers approximate KKT multipliers of (4), and hence $\nabla J(\tilde{y})$ represents an approximation of a Clarke subgradient. In practice, this approximation has not caused significant issues in our numerical experiments.

2.2 The feasibility auxiliary problem – $\mathcal{P}_{\text{FNLP}}$

For some $\hat{y} \in Y$, the problem $\mathcal{P}_{\text{NLP}}$ which we aim to solve might be infeasible. In this specific situation, we solve a feasibility problem to find the closest point to $\hat{y}$ that lies on the boundary of $\tilde{\mathcal{F}}$. The sought point is obtained by solving the following NLP

$$\begin{aligned} \min_{x \in X, y \in \tilde{Y}} \quad & \|y - \hat{y}\|_2^2 & (7a) \\ \text{s.t.} \quad & g(x, y) \leq 0, & (7b) \\ & h(x, y) = 0, & (7c) \\ & \|y_b - y\|_2^2 \leq \|y_b - \hat{y}\|_2^2, \text{ if } y_b \text{ given.} & (7d) \end{aligned}$$

$\mathcal{P}_{\text{FNLP}}(\hat{y}, y_b) :$

Constraint (7d) is enforced only if a feasible (best) solution $y_b \in \mathcal{F}$ is available. This constraint narrows the feasible set $\tilde{Y}$ by requiring that $\bar{y}$ lies within a ball of radius $\|y_b - \hat{y}\|_2$ around the best point y_b . This requirement is particularly helpful in case of a disconnected feasible set $\tilde{\mathcal{F}}$, but it does not introduce further complexity in case of a convex feasible set $\tilde{\mathcal{F}}$ (cf. Lemma 1).

Remark 2 By construction, $Y = \tilde{Y} \cap \mathbb{Z}^{n_y}$, and hence $Y \subseteq \tilde{Y}$. Consequently, the feasible set of the integer problem satisfies $\mathcal{F} \subseteq \tilde{\mathcal{F}}$, independently of convexity or regularity assumptions. In particular, if $\tilde{\mathcal{F}} \neq \emptyset$, then $\mathcal{P}_{\text{FNLP}}(\hat{y}, y_b)$ is feasible, since it reduces to the projection of $\hat{y}$ onto the nonempty set $\tilde{\mathcal{F}}$. If $\tilde{\mathcal{F}} = \emptyset$, then $\mathcal{P}_{\text{FNLP}}$ and hence the original MINLP are infeasible.

We denote by $\bar{y}(\hat{y}, y_b)$ the integer component of the solution of $\mathcal{P}_{\text{FNLP}}$. Moreover, we utilize $\bar{y}(\hat{y}, y_b)$ to construct the following infeasibility cut

$$(\hat{y} - \bar{y})^\top (y - \bar{y}) \leq 0. \quad (8)$$

This constraint can be interpreted as the hyperplane on a convex set going through $\bar{y}$ with a normal vector $\hat{y} - \bar{y}$. An illustration of the infeasibility cut is given in Fig. 2.

Lemma 1 Let $\hat{y} \in Y$ be such that $\hat{y} \notin \tilde{\mathcal{F}}$, and let $\bar{y}$ be a solution of $\mathcal{P}_{\text{FNLP}}(\hat{y}, y_b)$. Then $y = \hat{y}$ violates the infeasibility cut $(\hat{y} - \bar{y})^\top (y - \bar{y}) \leq 0$.

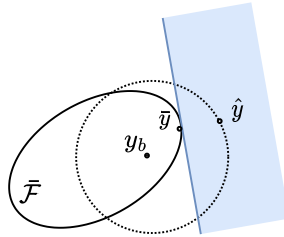


Fig. 2 In light blue an example of an infeasibility cut for a convex $\mathcal{P}_{\text{MINLP}}$, i.e., the relaxed feasible set $\tilde{\mathcal{F}}$ is a convex set and corresponds to the interior of the solid black line. $y_b \in \mathcal{F}$ is the current best and feasible solution, $\hat{y} \in \mathbb{Z}^{n_y}$ is an infeasible solution, and $\bar{y}$ is the solution of $\mathcal{P}_{\text{FNLP}}$

Proof By definition of $\mathcal{P}_{\text{FNLP}}(\hat{y}, y_b)$, the point $\bar{y}$ satisfies $\|\bar{y} - y_b\|_2^2 \leq \|\hat{y} - y_b\|_2^2$ whenever y_b is available. Moreover, since $\bar{y}$ minimizes $\|y - \hat{y}\|_2^2$ over the feasible set, we have $\bar{y} \neq \hat{y}$ whenever $\hat{y} \notin \tilde{\mathcal{F}}$. Thus, for $y = \hat{y}$, $(\hat{y} - \bar{y})^\top (\hat{y} - \bar{y}) = \|\hat{y} - \bar{y}\|_2^2 > 0$, which implies that $\hat{y}$ violates constraint (8). $\square$

Lemma 2 Under Assumption 3, let $\hat{y} \notin \tilde{\mathcal{F}}$, and let $\bar{y} \in \arg \min_{y \in \tilde{\mathcal{F}}} \|y - \hat{y}\|_2^2$. Then, the inequality $(\hat{y} - \bar{y})^\top (y - \bar{y}) \leq 0$ is valid for all $y \in \tilde{\mathcal{F}}$.

Proof Since $\bar{y}$ is the Euclidean projection of $\hat{y}$ onto the closed convex set $\tilde{\mathcal{F}}$, the first-order optimality condition of the projection problem states that, at the optimal point $\bar{y}$, the gradient $(y - \hat{y})$ evaluated at $\bar{y}$ must satisfy

$$(\hat{y} - \bar{y})^\top (y - \bar{y}) \leq 0 \quad \forall y \in \tilde{\mathcal{F}},$$

which is exactly the infeasibility cut (8). $\square$

Based on Lemma 2, we can conclude that the constraint (8) is satisfied for the best found solution y_b , if the problem is convex. On the contrary, for a nonconvex $\tilde{\mathcal{F}}$, we remark that constraints (7d) may be active. Thus, the corresponding infeasibility cut may not be tangential to $\tilde{\mathcal{F}}$.

2.3 Bookkeeping, lower bound function and benders region

Before introducing the two master problems, we present which data needs to be stored during algorithm iterations. Then, we show how these data are used for constructing J_{LB} , a function that outer approximates all the visited points and is relevant for the master problem $\mathcal{P}_{\text{LB-MILP}}$. Lastly, using J_{LB} we define the “Benders region” $\mathbb{B}$, which restricts the feasible set of $\mathcal{P}_{\text{BR-MIQP}}$ to a neighborhood of the current best solution, where we expect $\mathcal{P}_{\text{BR-MIQP}}$ to approximate $\mathcal{P}_{\text{MINLP}}$ well.

Consider a collection of points for which first-order information of J is available in terms of evaluation tuples $(i, x_i, y_i, J(y_i), \nabla J(y_i))$. We collect them in a dataset

$$\mathcal{D}_k := ((0, x_0, y_0, J(y_0), \nabla J(y_0)), \dots, (k, x_k, y_k, J(y_k), \nabla J(y_k))).$$

At each iteration k , we classify the outcome and store the results accordingly. If we successfully solve $\mathcal{P}_{\text{NLP}}$ and obtain a feasible solution with objective value $J(y_k)$, we add k to the index set $\mathbb{T}_k$, which tracks all feasible iterations. Otherwise, if we solve the feasibility problem $\mathcal{P}_{\text{FNLP}}(y_k, y_{b(k)})$ and obtain solution $\bar{y}_k$, we store this infeasible solution and add k to the index set $\mathbb{S}_k$. Note that $\mathbb{T}_k \cup \mathbb{S}_k = \mathbb{Z}_{[0,k]}$.

To track the best solution found up to iteration k , we define:

$$b(k) \in \begin{cases} \arg \min_{i \in \mathbb{T}_k} J(y_i), & \text{if } \mathbb{T}_k \neq \emptyset, \\ \arg \min_{i \in \mathbb{S}_k} \|\bar{y}_i - \hat{y}_i\|_2^2, & \text{otherwise.} \end{cases} \quad (9)$$

When multiple indices achieve the same minimum value, we select the smallest index to ensure uniqueness. Clearly, $b(k) \leq k$. If $b(k) \in \mathbb{T}_k$, then $(x_{b(k)}, y_{b(k)})$ represents the *incumbent* solution, i.e., the best feasible solution found so far, with objective value $\text{UB} = J(y_{b(k)})$.

The lower bound function J_{LB} is defined on the index sets $\mathbb{T}_k, \mathbb{S}_k$ and on the collected data $\mathcal{D}_k$ as follows

$$\begin{aligned} J_{\text{LB}}(y; \mathbb{T}_k, \mathbb{S}_k, \mathcal{D}_k) &:= \min_{\eta} \eta \\ \text{s.t. } \quad &\eta \geq J(y_i) + \nabla J(y_i)^\top (y - y_i), \quad i \in \mathbb{T}_k, \\ &(y_j - \bar{y}_j)^\top (y - \bar{y}_j) \leq 0, \quad j \in \mathbb{S}_k. \end{aligned} \quad (10)$$

The term $\nabla J(y_i)$ is a subgradient of the potentially nonsmooth function J , and its computation has been discussed in Sect. 2.1.

Consistent with the above discussion on restricting the search to a neighborhood of the incumbent solution, we define the Benders region at iteration k as

$$\mathbb{B}_k := \{y \in \mathbb{R}^{n_y} \mid J_{\text{LB}}(y; \mathbb{T}_k, \mathbb{S}_k, \mathcal{D}_k) \leq \bar{J}_k\}, \quad (11)$$

where $\bar{J}_k$ is determined by

$$\bar{J}_k := \alpha J(y_{b(k)}) + (1 - \alpha) \text{LB}, \quad \alpha \in [0, 1), \quad (12)$$

where α is a user-chosen parameter, and LB is the value of the best lower bound of MINLP (1) found by the algorithm. The computation of this reduced right-hand-side term follows an idea from [41]. It has the property to exclude all the visited points including the best point $y_{b(k)}$ from the current region $\mathbb{B}_k$. Therefore, a point $y \in \mathbb{B}_k$ must be outer approximated by the linear cuts constructed using previous solutions and must have an objective lower than $\bar{J}_k$.

2.4 Master problems: benders-region mixed-integer quadratic program –

$\mathcal{P}_{\text{BR-MIQP}}$

The first master problem we introduce is a MIQP since it can provide a more accurate approximation of $\mathcal{P}_{\text{MINLP}}$ compared to a mixed-integer linear program (MILP), ultimately requiring fewer outer approximation iterations to achieve a solution. Moreover, MIQP master problems have been shown to be favorable over MILP master problems in the context of outer approximation. In fact, it is easy to construct simple examples where outer approximation with MILP master problems requires complete enumeration to find the global optimum [28]. A disadvantage of MIQP master problems argued in [28] regards longer computation times compared to MILP. However, modern MIQP solvers have significantly improved in efficiency, narrowing the runtime gap between MILP and MIQP.

In particular, the “Benders-region MIQP”, $\mathcal{P}_{\text{BR-MIQP}}$, incorporates a quadratic approximation of $\mathcal{P}_{\text{MINLP}}$ computed around the current best point $(x_{b(k)}, y_{b(k)})$ with the Hessian approximation $B_{b(k)}$. The matrix $B_{b(k)}$ is a symmetric and positive semidefinite approximation of the Hessian of the Lagrangian of $\mathcal{P}_{\text{NLP}}$ with respect to (x, y) , evaluated at $(x_{b(k)}, y_{b(k)})$. In practice, $B_{b(k)}$ may correspond to the exact Hessian of the Lagrangian of $\mathcal{P}_{\text{MINLP}}$ evaluated using primal and dual information of the best solution found, or to quasi-Newton updates, or set to zero, yielding a first-order model. Problem $\mathcal{P}_{\text{BR-MIQP}}$ also includes additional constraints imposed by the Benders region (11) and infeasibility cuts (8). Hence, denoting the linearizations of functions f, g, h at $(\bar{x}, \bar{y})$ as $f_L(\cdot; \bar{x}, \bar{y})$, $g_L(\cdot; \bar{x}, \bar{y})$, and $h_L(\cdot; \bar{x}, \bar{y})$, respectively, we state

$\mathcal{P}_{\text{BR-MIQP}}$:

$$\begin{aligned}
 \min_{x \in X, y \in Y} \quad & f_L(x, y; x_{b(k)}, y_{b(k)}) + \frac{1}{2} \begin{pmatrix} x - x_{b(k)} \\ y - y_{b(k)} \end{pmatrix}^\top B_{b(k)} \begin{pmatrix} x - x_{b(k)} \\ y - y_{b(k)} \end{pmatrix} \\
 \text{s.t.} \quad & g_L(x, y; x_{b(k)}, y_{b(k)}) \leq 0, \\
 & h_L(x, y; x_{b(k)}, y_{b(k)}) = 0, \\
 & J(y_i) + \nabla J(y_i)^\top (y - y_i) \leq \bar{J}_k, \quad i \in \mathbb{T}_k, \\
 & (y_i - \bar{y}_i)^\top (y - \bar{y}_i) \leq 0, \quad i \in \mathbb{S}_k.
 \end{aligned} \tag{13}$$

We denote the objective value of (13) as V_{MIQP} . In the case no feasible solution has been found yet, i.e., $\mathbb{T}_k = \emptyset$, the functions f_L, g_L, h_L are obtained by linearizing at the infeasible point $(x_{b(k)}, y_{b(k)})$, where $b(k)$ is computed according to (9). Problem $\mathcal{P}_{\text{BR-MIQP}}$ depends on the convex polyhedron $\mathbb{B}_k$. There are no guarantees that an integer point y such that $y \in Y \cap \mathbb{B}_k$ exists, thus $\mathcal{P}_{\text{BR-MIQP}}$ might be infeasible. When this happens, we trigger the if-condition of line 22 and solve a different master problem, named $\mathcal{P}_{\text{LB-MILP}}$, which is presented in the next section.

2.5 Master problems: lowerbound MILP– $\mathcal{P}_{\text{LB-MILP}}$

We present the second master problem named “Lower-bound MILP”, $\mathcal{P}_{\text{LB-MILP}}$, which has a structure similar to the master problem of the GBD algorithm [31]. In our case, we additionally include in the constraints the linear approximation of the original MINLP (1) around the best solution found. This can be seen as a mix between GBD and linear outer approximation [28]. Problem $\mathcal{P}_{\text{LB-MILP}}$ is introduced because a quadratic outer approximation method cannot guarantee convergence to a global optimum even for convex MINLP as already shown in [28]. In fact, the solution of $\mathcal{P}_{\text{BR-MIQP}}$ (or an alternative MIQP master problem) does not provide an outer approximation of $\mathcal{P}_{\text{MINLP}}$. Specifically, solving $\mathcal{P}_{\text{LB-MILP}}$ lets us establish whether the current best solution is optimal or other solutions exist.

At the k -th iteration of the proposed algorithm, $\mathcal{P}_{\text{LB-MILP}}$ can be formulated as a MILP in the full variable space as follows

$\mathcal{P}_{\text{LB-MILP}}$:

$$\min_{\eta \in \mathbb{R}, x \in X, y \in Y} \eta \quad (14a)$$

$$\text{s.t.} \quad \eta \geq f_L(x, y; x_{b(k)}, y_{b(k)}), \quad \text{if } b(k) \in \mathbb{T}_k, \quad (14b)$$

$$0 \geq g_L(x, y; x_{b(k)}, y_{b(k)}), \quad \text{if } b(k) \in \mathbb{T}_k, \quad (14c)$$

$$0 = h_L(x, y; x_{b(k)}, y_{b(k)}), \quad \text{if } b(k) \in \mathbb{T}_k, \quad (14d)$$

$$\eta \geq J(y_i) + \nabla J(y_i)^\top (y - y_i), \quad i \in \mathbb{T}_k \setminus \{b(k)\}, \quad (14e)$$

$$0 \geq (y_i - \bar{y}_i)^\top (y - \bar{y}_i), \quad i \in \mathbb{S}_k. \quad (14f)$$

We denote the objective value of (14) as V_{MILP} . Unlike $\mathcal{P}_{\text{BR-MIQP}}$, the outer approximation objective cut (14b) and constraint cuts (14c)–(14d) are only imposed if the current best solution $(x_{b(k)}, y_{b(k)})$ is feasible, i.e., $b(k) \in \mathbb{T}_k$. This ensures that no cuts are imposed when the variables y are fractional. In $\mathcal{P}_{\text{LB-MILP}}$, for each feasible visited solution y_i we construct a linear approximator in the integer space of the nonlinear function $J : Y \mapsto \mathbb{R}$, cf., (2). Thus, by minimizing the slack variable η , we aim to find the minimum of the epigraph of the function obtained by the intersection of all the epigraphs of linear models. Since we impose the OA constraints (14b)–(14d), the Benders constraints (14e) are imposed for all the indices in $\mathbb{T}_k$ but $b(k)$ if $b(k) \in \mathbb{T}_k$. Note that the set minus operation is defined such that if $b(k) \notin \mathbb{T}_k$ then $\mathbb{T}_k \setminus \{b(k)\} = \mathbb{T}_k$. Under Assumption 3, the objective value V_{MILP} of $\mathcal{P}_{\text{LB-MILP}}$ at an optimal solution provides a lower bound on the objective value of $\mathcal{P}_{\text{MINLP}}$. Therefore, the lower bound LB is updated whenever the solution of $\mathcal{P}_{\text{LB-MILP}}$ yields a V_{MILP} higher than the current LB. Eventually, in case $\mathcal{P}_{\text{LB-MILP}}$ is infeasible we set $V_{\text{MILP}} = +\infty$. Infeasibility of $\mathcal{P}_{\text{LB-MILP}}$ might happen if at the iteration k the algorithm has tested every value $y \in Y$ and $\mathbb{T}_k = \emptyset$, for further details see Sect. 2.7.

2.6 The S-BMIQP algorithm

Algorithm 1 presents the complete Sequential Benders MIQP (S-B-MIQP) algorithm in detail. First, in this subsection, we consider Assumption 3 about convexity to hold; thus, line 15 is omitted, and we assume $\tilde{\mathcal{D}}_k := \mathcal{D}_k$. We address the nonconvex case in Sect. 3.

Consistent with the notation adopted earlier, in Algorithm 1, the letter k denotes the iteration index, where $k \in \mathbb{Z}$. Algorithm 1 begins with an integer point $y_0 \in Y$ provided by the user, along with a lower bound, LB, obtained by solving the integer relaxation of (1), while the upper bound is $UB = +\infty$. Following initialization, we enter a cycle that terminates only when the lower bound becomes equal to the upper bound, UB. This termination condition is common for mixed-integer programming algorithms.

The first half of the cycle involves evaluating the quality of the integer solution y_k and solving for the continuous variable x . In each iteration, the upper bound is updated by comparing the current UB with $J(y_k)$, the objective value of $\mathcal{P}_{\text{NLP}}$. In case $\mathcal{P}_{\text{NLP}}$ is infeasible for a given y_k , we solve the feasibility problem $\mathcal{P}_{\text{FNLP}}$, whose solution is used to construct specific cutting planes aiming to steer the integer solution back to the feasible set. The information obtained in each iteration is stored in $\mathcal{D}_k$. The index sets $\mathbb{T}_k$ and $\mathbb{S}_k$ are updated based on the feasibility of $\mathcal{P}_{\text{NLP}}$.

The second half of Algorithm 1 focuses on computing new integer solutions and lower bounds. The integer solutions are computed in the master problems $\mathcal{P}_{\text{BR-MIQP}}$ and $\mathcal{P}_{\text{LB-MILP}}$. The idea is to approach the minimizer of $\mathcal{P}_{\text{MINLP}}$ (1) by solving MIQPs, as done in SQP for continuous problems. Specifically, $\mathcal{P}_{\text{BR-MIQP}}$ is solved if a new incumbent solution is found in the current or in the previous iteration, cf. line 17 of Algorithm 1. In case, the incumbent solution is not improving or $\mathcal{P}_{\text{BR-MIQP}}$ becomes infeasible due to tighter Benders regions, Algorithm 1 switches to solving $\mathcal{P}_{\text{LB-MILP}}$. The solution of $\mathcal{P}_{\text{LB-MILP}}$ provides a valid lower bound on the objective of $\mathcal{P}_{\text{MINLP}}$ (1). The solution of $\mathcal{P}_{\text{BR-MIQP}}$ is resumed only if the integer solution found by $\mathcal{P}_{\text{LB-MILP}}$ produces a new incumbent solution. In this case, Algorithm 1 attempts to solve a new $\mathcal{P}_{\text{BR-MIQP}}$ constructed around the new incumbent solution.

2.7 Algorithm properties

Before stating the theoretical properties of Algorithm 1, we emphasize that the proofs rely on standard results from convex analysis and outer-approximation theory. In particular, Assumption 3 guarantees that first-order Taylor expansions yield global underestimators, while Assumption 2 ensures existence and boundedness of Lagrange multipliers and validity of first-order optimality conditions.

Lemma 3 *A continuously differentiable function $f : X \rightarrow \mathbb{R}$ is convex if and only if X is a convex set and $f(z) \geq f(x) + \nabla f(x)^\top (z - x)$ holds for all $x, z \in X$.*

Proof Given in [12, §3.1.3]. □

Algorithm 1 Sequential Benders MIQP (S-B-MIQP)

```

1: Initialize:  $y_0 \in Y, \mathbb{B}_0 \leftarrow Y, k \leftarrow 0, \mathbb{S}_{-1} \leftarrow \emptyset, \mathbb{T}_{-1} \leftarrow \emptyset, \text{UB} = +\infty, \text{LB} = -\infty,$ 
    $b(k) \leftarrow 0, \text{needMILP} \leftarrow \text{False}$ 
2:  $\text{LB} \leftarrow \min_{y \in \tilde{Y}} J(y)$   $\triangleright$  LB given by the objective of MINLP relaxation
3: while  $\text{LB} < \text{UB}$  do:
4:   Given  $y_k$  solve  $\mathcal{P}_{\text{NLP}}$ 
5:   if  $\mathcal{P}_{\text{NLP}}$  is feasible then
6:     Store solution  $(k, x_k, y_k, J(y_k), \nabla J(y_k))$  in  $\mathcal{D}_k$  and  $\mathbb{T}_k \leftarrow \mathbb{T}_{k-1} \cup \{k\}$ 
7:     if  $J(y_k) < \text{UB}$  then
8:        $\text{UB} \leftarrow J(y_k)$   $\triangleright$  Update upper bound
9:     end if
10:   else
11:     Solve  $\mathcal{P}_{\text{FNLP}}(y_k, y_{b(k)})$  and get  $\tilde{y}_k$ 
12:     Store solution  $(k, x_k, y_k, \tilde{y}_k, \|\tilde{y}_k - y_k\|_2^2)$  in  $\mathcal{D}_k$  and  $\mathbb{S}_k \leftarrow \mathbb{S}_{k-1} \cup \{k\}$ 
13:   end if
14:   Compute  $b(k)$  according to (9)
15:   Modify gradients in  $\mathcal{D}_k$ , get  $\tilde{\mathcal{D}}_k$   $\triangleright$  Required only for noncvx MINLP (cf.
      Sect. 3)
16:   Compute Benders region  $\mathbb{B}_k$  based on  $\tilde{\mathcal{D}}_k$  according to (11)
17:   if  $k - b(k) \leq 1$  then  $\triangleright$  Best solution found in the last iteration
18:     Solve  $\mathcal{P}_{\text{BR-MIQP}}$  given  $(x_{b(k)}, y_{b(k)}), \mathbb{B}_{b(k)}$ , and  $\mathbb{B}_k$ . Get its solution  $\tilde{y}$ .
19:   else
20:      $\text{needMILP} \leftarrow \text{True}$ 
21:   end if
22:   if  $\text{needMILP}$  is True or  $\mathcal{P}_{\text{BR-MIQP}}$  is infeasible then:
23:     Solve  $\mathcal{P}_{\text{LB-MILP}}$  with  $J_{\text{LB}}(y; \mathbb{T}_k, \mathbb{S}_k, \tilde{\mathcal{D}}_k)$ , get solution  $\tilde{y}$  and  $V_{\text{MILP}}$ 
24:     if  $\mathcal{P}_{\text{LB-MILP}}$  is infeasible then
25:        $\text{LB} \leftarrow +\infty$ 
26:     else
27:       if  $V_{\text{MILP}} > \text{LB}$  then
28:          $\text{LB} \leftarrow V_{\text{MILP}}$   $\triangleright$  Update lower bound
29:       end if
30:     end if
31:      $\text{needMILP} \leftarrow \text{False}$ 
32:   end if
33:    $y_{k+1} \leftarrow \tilde{y}$ 
34:    $b(k+1) \leftarrow b(k)$ 
35:    $k \leftarrow k+1$ 
36: end while

```

Lemma 4 Suppose Assumptions 1 and 3 hold. Assume further that all integer points have been visited at iteration k , i.e., $Y = \{y_i \mid i \in \mathbb{T}_k \cup \mathbb{S}_k\}$. Then the lower-bound master problem $\mathcal{P}_{\text{LB-MILP}}$ has the same set of optimal integer solutions as $\mathcal{P}_{\text{MINLP}}$.

Proof Proved in [43, §13.1] for GBD. Here, the only difference is that the master problem $\mathcal{P}_{\text{LB-MILP}}$ imposes OA constraints about $(x_{b(k)}, y_{b(k)})$ if $b(k) \in \mathbb{T}_k$, while in the standard GBD master problem we would have a Benders cut like (14e) also for $(x_{b(k)}, y_{b(k)})$. Thus, the feasible set of $\mathcal{P}_{\text{LB-MILP}}$ is tighter compared to the one of the standard master problem in GBD. Below, we provide the logical steps of the proof.

Under Assumption 3, the functions f and $g_1, \dots, g_{n_g}$ are convex and h is affine on $X \times \bar{Y}$. Hence, by Lemma 3, their first-order Taylor expansions define global underestimators on $X \times \bar{Y}$. Therefore, every feasible solution to $\mathcal{P}_{\text{MINLP}}$ is feasible for $\mathcal{P}_{\text{LB-MILP}}$. This shows that $\mathcal{P}_{\text{LB-MILP}}$ is an outer approximation of $\mathcal{P}_{\text{MINLP}}$. Moreover, for each feasible integer point $y_i \in \mathbb{T}_k$, the constraint $\eta \geq J(y_i) + \nabla J(y_i)^\top (y - y_i)$ is included in $\mathcal{P}_{\text{LB-MILP}}$. Since all feasible integer points have been visited, these inequalities describe the epigraph of J over $\mathcal{F}$ exactly. For each infeasible integer point $y_i \in \mathbb{S}_k$, the infeasibility cut $(y_i - \bar{y}_i)^\top (y - \bar{y}_i) \leq 0$ excludes y_i from the feasible region of $\mathcal{P}_{\text{LB-MILP}}$. Hence, the feasible integer assignments of $\mathcal{P}_{\text{LB-MILP}}$ coincide exactly with $\mathcal{F}$. Consequently, minimizing η over $\mathcal{P}_{\text{LB-MILP}}$ is equivalent to minimizing $J(y)$ over Y . Therefore, the two problems share the same set of optimal integer solutions. $\square$

Theorem 1 If Assumptions 1, 2 and 3 hold, Algorithm 1 terminates in a finite number of iterations and returns either a globally optimal solution of $\mathcal{P}_{\text{MINLP}}$ or a certificate of infeasibility.

Proof We first show that no integer assignment is repeated. If $\mathcal{P}_{\text{NLP}}$ is infeasible at y_k , then the infeasibility cut $(y_k - \bar{y}_k)^\top (y - \bar{y}_k) \leq 0$ excludes y_k from subsequent master problems. If $\mathcal{P}_{\text{NLP}}$ is feasible and $\mathcal{P}_{\text{BR-MIQP}}$ returns a new integer solution, then by construction $\tilde{y} \neq y_k$. If $\mathcal{P}_{\text{BR-MIQP}}$ is infeasible or stagnates, the algorithm solves $\mathcal{P}_{\text{LB-MILP}}$. If $\mathcal{P}_{\text{LB-MILP}}$ returns $\tilde{y} = y_k$, then the stopping condition $\text{LB} \geq \text{UB}$ is satisfied and the algorithm terminates. Hence, no integer assignment is revisited. Since Y is finite by Assumption 1, finite termination follows.

We now prove correctness under convexity. Under Assumption 3 and Lemma 3, the linearizations of f , g , and h are global underestimators. Therefore, $\mathcal{P}_{\text{LB-MILP}}$ is an outer approximation of $\mathcal{P}_{\text{MINLP}}$. Consequently, every feasible solution to $\mathcal{P}_{\text{MINLP}}$ remains feasible for $\mathcal{P}_{\text{LB-MILP}}$. Let (x^*, y^*) be the globally optimal solution of $\mathcal{P}_{\text{MINLP}}$ with objective f^* , hence $\mathcal{P}_{\text{NLP}}$ is feasible for $J(y^*)$ and $\text{UB} = J(y^*) = f(x^*, y^*)$. Since (x^*, y^*) is feasible for $\mathcal{P}_{\text{MINLP}}$, it is also feasible for $\mathcal{P}_{\text{LB-MILP}}$ by the outer approximation property. Evaluating the epigraph constraints of $\mathcal{P}_{\text{LB-MILP}}$ at (x^*, y^*) yields

$$\eta \geq f(x^*, y^*) + \nabla f(x^*, y^*)^\top \begin{pmatrix} \tilde{x} - x^* \\ y^* - y^* \end{pmatrix}, \quad (15a)$$

$$\eta \geq J(y_i) + \nabla J(y_i)^\top (y^* - y_i), \quad i \in \mathbb{T}_k \setminus \{b(k)\}. \quad (15b)$$

From (15a), we know that no valid descent direction exists along x at x^* (cf. Assumption 2), therefore

$$\nabla f(x^*, y^*)^\top \begin{pmatrix} \tilde{x} - x^* \\ 0 \end{pmatrix} \geq 0.$$

By convexity of J , each Benders cut (15b) satisfies

$$J(y_i) + \nabla J(y_i)^\top (y^* - y_i) \leq J(y^*) = f(x^*, y^*) = \text{UB}, \quad \text{for all } i \in \mathbb{T}_k \setminus \{b(k)\},$$

it follows

$$\text{LB} = \eta \geq f(x^*, y^*) + \nabla f(x^*, y^*)^\top \begin{pmatrix} \tilde{x} - x^* \\ 0 \end{pmatrix} \geq \text{UB},$$

thus Algorithm 1 terminates.

Assume Algorithm 1 terminates at a point (x', y') which is not the global optimum, hence with objective value $J(y')$ strictly greater than $f(x^*, y^*)$. Termination implies $\text{LB} \geq \text{UB}$. Since $\text{UB} = J(y')$, we have $\text{LB} \geq J(y') > f(x^*, y^*)$. However, (x^*, y^*) is feasible for $\mathcal{P}_{\text{LB-MILP}}$, so $\text{LB} \leq f(x^*, y^*)$. This is a contradiction.

Algorithm 1 classifies $\mathcal{P}_{\text{MINLP}}$ as infeasible when every $y \in Y$ renders $\mathcal{P}_{\text{NLP}}$ infeasible. By definition of $\mathcal{F}$, this implies $\mathcal{F} = \emptyset$. Hence, $\mathcal{P}_{\text{MINLP}}$ is infeasible. Moreover, for each $y \in Y$ an infeasibility cut excludes y from the feasible set of $\mathcal{P}_{\text{LB-MILP}}$. Since Y is finite, after all integer points have been processed, no feasible integer assignment remains. Therefore, $\mathcal{P}_{\text{LB-MILP}}$ becomes infeasible and the Algorithm 1 terminates with $\text{LB} = \text{UB} = +\infty$. $\square$

2.8 Tutorial example for S-B-MIQP

We illustrate the behavior of Algorithm 1 for a MINLP where we can have an effective graphical representation. The example is taken from [32]. Consider the following convex MINLP

$$\min_{\substack{x \in \mathbb{R}, \\ (y_{[1]}, y_{[2]}) \in \mathbb{Z}^2}} (y_{[1]} - 4.1)^2 + (y_{[2]} - 4.0)^2 + \lambda x \quad (16a)$$

$$\text{s.t.} \quad y_{[1]}^2 + y_{[2]}^2 - r^2 - x \leq 0, \quad (16b)$$

$$-x \leq 0, \quad (16c)$$

where $r = 3$ and $\lambda = 1000$. The term λx in (16a) can be seen as a penalization of the violation of the quadratic constraint $y_{[1]}^2 + y_{[2]}^2 - r^2 \leq 0$. The shape of the cost function and the global minimizer of (16) are represented in Fig. 3. The global minimizer can be found graphically and corresponds to $(x, y_{[1]}, y_{[2]}) = (0, 2, 2)$. Note that x is determined by $x = \max(0, y_{[1]}^2 + y_{[2]}^2 - r^2)$.

Given a linearization point $(\bar{x}, \bar{y}_{[1]}, \bar{y}_{[2]})$, one can compute the corresponding master problems, $\mathcal{P}_{\text{BR-MIQP}}$ and $\mathcal{P}_{\text{LB-MILP}}$. Since the original cost is convex and quadratic, one can choose B as the exact Hessian of (16a). The quadratic constraint (16b) is

Fig. 3 Level lines of the cost function of problem (16). The red asterisk denotes the global minimizer

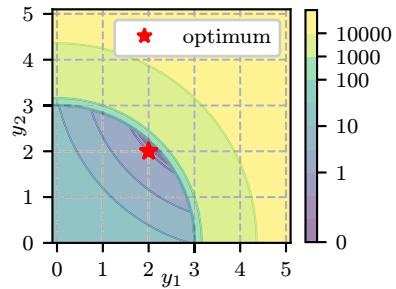


Table 1 Iterations of Algorithm 1 for problem (16)

k	LB	UB	$b(k)$	y_k	$J(y_k)$	V_k
0	7.44	7016.81	0	(0, 4)	7016.81	-
1	7.44	7016.81	0	(4, 3)	16001.01	1.01
2	7.44	4005.21	2	(3, 2)	4005.21	5.21
3	8.41	8.41	3	(2, 2)	8.41	8.41

linearized to fit the MILP/MIQP approximation as follows

$$-\bar{y}_{[1]}^2 - \bar{y}_{[2]}^2 + 2y_{[1]}\bar{y}_{[1]} + 2y_{[2]}\bar{y}_{[2]} - x \leq r^2. \quad (17)$$

Also, for the MILP/MIQP approximation x is implicitly defined as $x = \max(0, -\bar{y}_{[1]}^2 - \bar{y}_{[2]}^2 + 2y_{[1]}\bar{y}_{[1]} + 2y_{[2]}\bar{y}_{[2]} - r^2)$. For this reason, in the following, we focus only on the values of $y_{[1]}, y_{[2]}$. We stack $y_{[1]}, y_{[2]}$ in the vector y_k as $y_k := (y_{[1]}, y_{[2]})$ where the subscript k denotes the iteration number of Algorithm 1.

We apply Algorithm 1 to solve (16), with hyperparameter $\alpha = 0.9$ for the reduced right-hand-side (12). Table 1 reports the results of each iteration, and Fig. 4 gives a graphical interpretation.

The problem is initialized with $y_0 = (0, 4)$, and the lower bound computed via MINLP relaxation is $LB = 7.44$. In the first iteration, the initial point y_0 is evaluated by solving $J(y_0)$. Since the problem is feasible, its objective value updates the initial upper bound, $UB = 7016.81$. Then, we compute the Benders region $\mathbb{B}_0$, and we solve $\mathcal{P}_{BR-MIQP}$ with linearization point (x_0, y_0) and Hessian B_0 . Since $\mathcal{P}_{BR-MIQP}$ is feasible, we store its objective in V_1 and its solution in $y_{[1]}$, hence $V_1 = 1.01$ and $y_{[1]} = (4, 3)$. The next two iterations of Algorithm 1, i.e., $k = 1, 2$ have a similar structure to the first iteration. We move our focus to the fourth iteration where $y_3 = (2, 2)$, and the associated $J(y_3)$ objective is 8.41, then UB is updated. Also, the Benders region $\mathbb{B}_3$ is computed, and its intersection with the integer feasible set is empty. Therefore, we switch to the solution of the associated $\mathcal{P}_{LB-MILP}$, whose minimizer is $y_{[2]} = (2, 2)$, and the objective is $V_{MILP} = 8.41$. The lower bound LB is updated and, finally, the termination condition of Algorithm 1 is met since $UB = LB$.

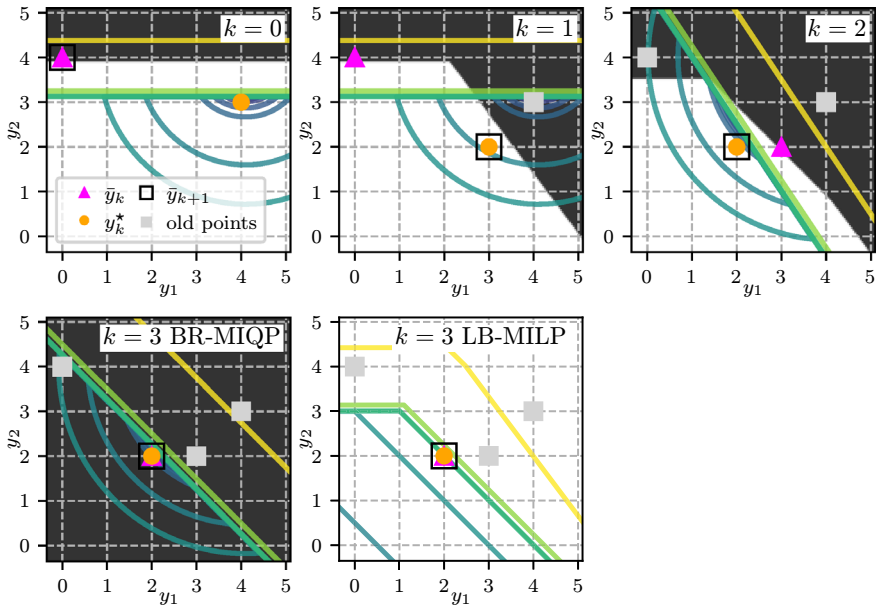


Fig. 4 Representation of algorithm iterations. The level lines correspond to either $\mathcal{P}_{\text{BR-MIQP}}$ or $\mathcal{P}_{\text{LB-MILP}}$, according to the iteration. The darkened areas describe the regions excluded by the Benders regions $\mathbb{B}_k$

3 Extension to the nonconvex case

Under the convexity Assumption 3, Theorem 1 guarantees termination of Algorithm 1 either to the globally optimal solution of the convex MINLP or with a certificate of infeasibility. However, when we deal with nonconvex MINLPs (1), for which we only require Assumptions 1 and 2 to hold, the termination condition of Algorithm 1 might be triggered in unintended situations. Due to nonconvexities, the cutting planes based on J_{LB} are not guaranteed to be lower bounds for the global optimum or for the current best solution.

We delineate two premature termination scenarios. The first scenario occurs when the solution of $\mathcal{P}_{\text{LB-MILP}}$ yields a new point (x_k, y_k) with a value exceeding the current UB. This situation is inherently ambiguous because we halt the algorithm due to a newfound minimizer that we acknowledge to be inferior to the best point found during the algorithm's iterations. The second scenario arises from infeasibility cutting planes, causing $\mathcal{P}_{\text{LB-MILP}}$ to become infeasible. Here, we set LB to $+\infty$, triggering Algorithm 1 termination. This second scenario is particularly misleading as it encroaches upon the condition $\text{LB} = +\infty$, typically reserved for detecting infeasibility in the original problem $\mathcal{P}_{\text{MINLP}}$.

We illustrate the first situation of premature termination with the following example. Consider the nonlinear integer program

$$\min_{y \in \mathbb{Z}_{[-4,4]}} (y^2 - 5)^2 + 4y. \quad (18)$$

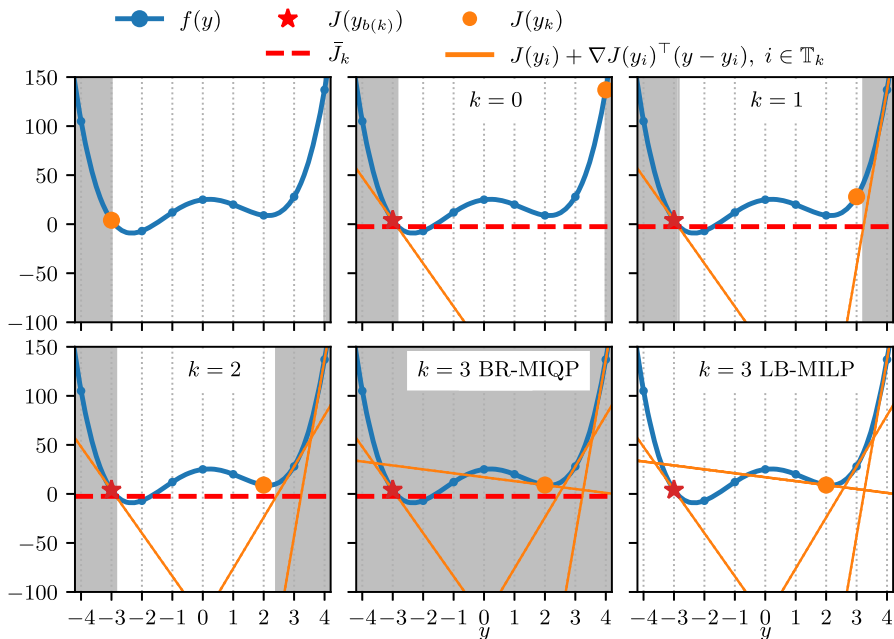


Fig. 5 Iterations of Algorithm 1 for Problem (18) with LB equal to the global minimum and $\alpha = 0.5$, starting from $y_0 = -3$. The shaded dark areas correspond to the area excluded from the feasible set by the given Benders region $\mathbb{B}_k$. The first plot depicts the initial condition of Algorithm 1, and the last plot illustrates its termination, highlighting the first scenario of premature termination where the solution of $\mathcal{P}_{\text{LB-MILP}}$ has value higher than the current UB

For simplicity, we run Algorithm 1 with zero Hessian $B = 0$ in $\mathcal{P}_{\text{BR-MIQP}}$, resulting in an MILP. As shown in Fig. 5, at the last iteration, the new Benders region $\mathbb{B}_3$ is empty. Hence, we attempt to solve $\mathcal{P}_{\text{LB-MILP}}$. Note that for the latter problem, the cutting planes are not lower bounds of the current best solution $y_{b(3)} = -3$. The solution of $\mathcal{P}_{\text{LB-MILP}}$ is $y = 2$, which has a value greater than UB, triggering the termination of Algorithm 1.

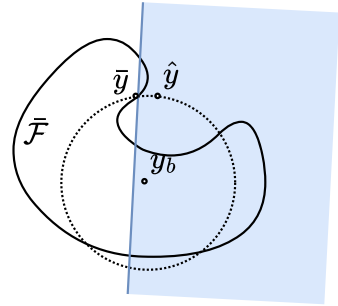
The second scenario is illustrated in Fig. 6, where $\tilde{\mathcal{F}}$ is nonconvex, $y_b \in \tilde{\mathcal{F}}$ is the best solution found, and $\hat{y}$ is the infeasible solution that Algorithm 1 has computed lastly. In the attempt to create an infeasibility cut as (8) for $\mathcal{P}_{\text{FNLP}}(\hat{y}, y_b)$, we cut a portion of the feasible set that includes y_b . This makes $\text{LB} = \infty$, triggering the termination of Algorithm 1.

In what follows, we introduce heuristic safeguarding strategies that aim to prevent premature termination in the nonconvex case, without restoring the global validity guarantees enjoyed under convexity. First, we present a procedure to correct the gradient of the linear models, then we show a way to enlarge the Benders region.

3.1 Gradient correction

During the computation of the underestimator J_{LB} at the k -th iteration of Algorithm 1, linearizations of the value function are constructed using vectors $\nabla J(y_i) \in \partial_C J(y_i)$

Fig. 6 Termination condition triggered by infeasibility cuts. The relaxed integer feasible set $\bar{\mathcal{F}}$ corresponds to the interior of the solid black line, $y_b \in \mathcal{F}$ is the current best and feasible solution, $\hat{y} \in \mathbb{Z}^{n_y}$ is infeasible, and $\bar{y}$ is the solution of (7). The infeasibility cut in $\bar{y}$ makes y_b infeasible and triggers the termination condition



obtained from auxiliary problems at points $y_i, i \in \mathbb{T}_k$. In contrast to the convex case, Clarke subgradients of the generally nonconvex value function J provide only local first-order sensitivity information and do not, in general, define globally valid Benders cuts. As a consequence, the linearization induced by $\nabla J(y_i)$ may overestimate the value function at other points, including the current best incumbent solution $y_{b(k)}$. Specifically, it may occur that for some $i \in \mathbb{T}_k$,

$$J(y_i) + \nabla J(y_i)^\top (y_{b(k)} - y_i) > J(y_{b(k)}), \quad (19)$$

in which case the corresponding linearization is not a valid Benders cut and may prematurely terminate Algorithm 1.

To ensure that the underestimator remains valid, we replace such local linearizations by corrected cutting planes that underestimate the value function at the incumbent solution. Specifically, we require the cutting plane associated with y_i to satisfy

$$J(y_i) + \tilde{g}^\top (y_{b(k)} - y_i) \leq J(y_{b(k)}), \quad (20)$$

where $\tilde{g}$ denotes a cut-generating vector.

The set of admissible cut-generating vectors is therefore defined as

$$\mathbb{G}_{(i,k)} := \left\{ \tilde{g} \mid J(y_i) + \tilde{g}^\top (y_{b(k)} - y_i) \leq J(y_{b(k)}) \right\}. \quad (21)$$

Elements of $\mathbb{G}_{(i,k)}$ are not required to belong to the Clarke subdifferential of J .

Among all admissible cut-generating vectors, we select the one that minimally deviates from the local sensitivity vector $\nabla J(y_i)$ in a weighted Euclidean norm with symmetric positive definite weight matrix W . This leads to the optimization problem

$$g_{(i,k)}^{\text{corr}} \in \arg \min_{g \in \mathbb{G}_{(i,k)}} \frac{1}{2} \|g - \nabla J(y_i)\|_W^2. \quad (22)$$

By default, the identity matrix is used as weight matrix W .

Lemma 5 *Problem (22) admits a closed-form solution. Let $\Delta y_{(i,k)} = y_{b(k)} - y_i$. Let $r_{(i,k)} = J(y_{b(k)}) - J(y_i) - \nabla J(y_i)^\top \Delta y_{(i,k)}$. Then the corrected cut-generating vector*

is given by

$$g_{(i,k)}^{\text{corr}} = \nabla J(y_i) + \begin{cases} 0, & \text{if } r_{(i,k)} \geq 0, \\ \frac{r_{(i,k)}}{\Delta y_{(i,k)}^\top W^{-1} \Delta y_{(i,k)}} W^{-1} \Delta y_{(i,k)}, & \text{if } r_{(i,k)} < 0. \end{cases} \quad (23)$$

Proof Let us write problem (22) in the following form

$$\begin{aligned} \min_{\Delta g} \quad & \frac{1}{2} \|\Delta g\|_W^2 \\ \text{s.t.} \quad & \Delta g^\top \Delta y - r \leq 0, \end{aligned} \quad (24)$$

where $\Delta g = g_{(i,k)}^{\text{corr}} - \nabla J(y_i)$, $\Delta y = y_{b(k)} - y_i$, and $r = J(y_{b(k)}) - J(y_i) - \nabla J(y_i)^\top \Delta y$. The Lagrangian function of the problem is given by

$$\mathcal{L}(\Delta g, \lambda) = \frac{1}{2} \|\Delta g\|_W^2 + (\Delta g^\top \Delta y - r)\lambda, \quad (25)$$

and the KKT system is given by

$$\begin{cases} W\Delta g + \lambda\Delta y & = 0, \\ \Delta g^\top \Delta y - r & \leq 0, \\ \lambda & \geq 0, \\ \lambda(\Delta g^\top \Delta y - r) & = 0. \end{cases} \quad (26)$$

In order to solve the system, we distinguish two cases:

1. If the constraint is inactive or weakly active $\Delta g^\top \Delta y - r \leq 0$ and $\lambda = 0$. The optimal solution can be directly obtained from the stationarity condition and $\Delta g = 0$.
2. If the constraint is strictly active $\Delta g^\top \Delta y - r = 0$ and $\lambda > 0$. From the stationarity condition we obtain $\Delta g = -\lambda W^{-1} \Delta y$. Substituting Δg into the primal feasibility condition we get $\lambda = -\frac{r}{\Delta y^\top W^{-1} \Delta y}$. By substitution, we find

$$\Delta g = \frac{r}{\Delta y^\top W^{-1} \Delta y} W^{-1} \Delta y.$$

□

When the best point does not change in the current k -th iteration, we only need to check if the new point y_k verifies inequality (20) with $\tilde{g} \equiv \nabla J(y_k)$ at the incumbent solution. If necessary, we correct its gradient. However, when y_k is the new best point, we must verify if inequality (20) holds for all points y_i , $i \in \mathbb{T}_k$ stored in $\mathcal{D}_k$ and correct the problematic gradients.

Remark 3 The correction step deliberately sacrifices the subgradient interpretation in order to guarantee global validity of the Benders underestimator at the incumbent solution. In the convex case, where subgradients yield globally valid cuts, the correction

is inactive. In the nonconvex setting, the procedure can be interpreted as a projection of a local sensitivity vector onto the set of admissible Benders cuts for the incumbent solution.

We denote by $\mathcal{D}_k^{\text{corr}}$ the “corrected” dataset at iterate k , which contains the corrected cut-generating vectors $g_{(i,k)}^{\text{corr}}$ instead of the local sensitivity vectors $\nabla J(y_i)$.

Lemma 6 *Under Assumptions 1 and 3, the corrected gradients equal the original ones, i.e., $\mathcal{D}_k^{\text{corr}} = \mathcal{D}_k$.*

Proof This directly follows from Lemma 3. Since J is a convex function with convex domain Y , the Clarke subdifferential coincides with the convex subdifferential. Hence, it holds $J(y_i) + \nabla J(y_i)^\top (y_j - y_i) \leq J(y_j)$, for any $y_i, y_j \in Y$. $\square$

3.2 Region expansion via gradient amplification

When the gradient correction is computed in a nonconvex situation, we find the minimum correction that ensures that the best point found is outer approximated by every cut, which potentially makes such point the only one feasible for $\mathcal{P}_{\text{LB-MILP}}$. The gradient correction resolves the ambiguous termination described at the beginning of this section, but it can dramatically limit Algorithm 1 from further exploring the integer solution space. For this purpose, we introduce a constant value $\rho \geq 1$, which amplifies all the gradients of the available linear model as follows:

$$g_{(i,k,\rho)}^{\text{ampl}} := \rho g_{(i,k)}^{\text{corr}}. \quad (27)$$

The amplification factor ρ is a hyperparameter of Algorithm 1, which could, for example, be chosen offline and kept fixed at runtime. More elaborate strategies to choose ρ separately per inequality and iteration index are also possible but are beyond our interest in this work.

We denote by $\mathcal{D}_k^{\text{ampl}}$ the dataset where the corrected and amplified gradients replace the original gradients. This set is used as the modified dataset in line 15 of Algorithm 1, i.e., the final version of Algorithm 1 sets $\tilde{\mathcal{D}}_k := \mathcal{D}_k^{\text{ampl}}$.

In Fig. 7 we illustrate how the combination of gradient correction and gradient amplification can solve a deadlock situation caused by nonconvexity.

Remark 4 We amplify by factor ρ only the gradients that are corrected. From this, it follows that for convex MINLPs, there is no amplification, i.e., $\rho = 1$, since there is no gradient correction, cf., Lemma 6.

At the end of this section, we demonstrate that incorporating gradient correction and amplification does not compromise the properties of Algorithm 1, namely, its finiteness and convergence to a global minimum under Assumption 3.

3.3 Correction of the infeasibility cuts

As shown in Fig. 6, the infeasibility cuts may cause early termination of Algorithm 1 by excluding the current feasible and best solution. To prevent this situation, we perform a

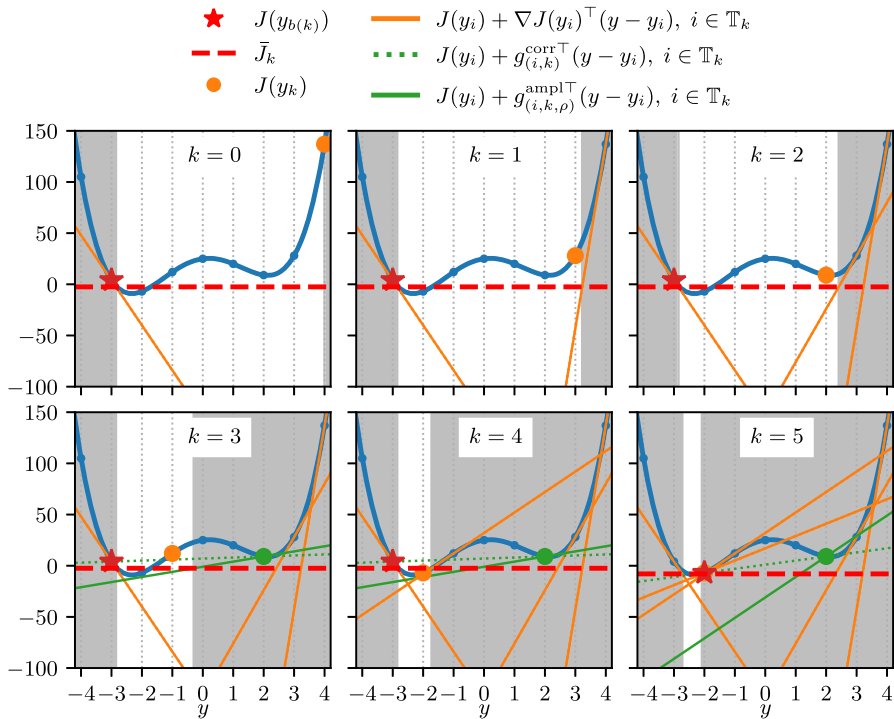


Fig. 7 We consider the same problem depicted in Fig. 5 and same settings for Algorithm 1, i.e., hyperparameter $\alpha = 0.5$. Here, we overcome the deadlock via the gradient correction and amplification, choosing $\rho = 5$. The gray area corresponds to infeasible values for the Benders region $\mathbb{B}_k$ constraint

cut correction similar to the one introduced in Sect. 3.1. Again, we modify the normal vector of the infeasibility cut such that at iteration k the following inequality holds

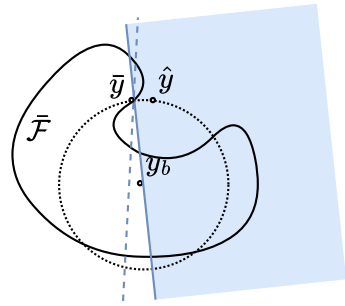
$$\tilde{n}_{(i,k)}^\top (y_{b(k)} - \bar{y}_i) \leq 0, \quad i \in \mathbb{S}_k, \quad (28)$$

where the corrected normal vector $\tilde{n}_{(i,k)}$ corresponds to the minimal correction of the original cut normal in a weighted Euclidean norm, similarly to (22). Hence,

$$n_{(i,k)}^{\text{corr}} \in \arg \min_{n \in \mathbb{N}_{(i,k)}} \frac{1}{2} \|n - (\hat{y}_i - \bar{y}_i)\|_W^2, \quad (29)$$

where $W \succ 0$ and $\mathbb{N}_{(i,k)} := \{\tilde{n} \mid \tilde{n}^\top (y_{b(k)} - \bar{y}_i) \leq 0\}$. We store the corrected cut normals $n_{(i,k)}^{\text{corr}}$ in the dataset $\mathcal{D}_k^{\text{corr}}$. We emphasize that we correct the infeasibility cuts exclusively when Algorithm 1 has found at least one feasible solution $y_{b(k)} \in \mathcal{F}$. In case no feasible solution is found, we enforce the infeasibility cuts as defined in (8). Fig. 8 illustrates the correction of the infeasibility cut for the scenario depicted earlier in Fig. 6.

Fig. 8 Correction of the infeasibility cut of Fig. 6 such that y_b remains feasible. The corrected cut is depicted solid blue, while the original cut is dashed blue



Lemma 7 Under Assumptions 1 and 3, the corrected cut normals of the infeasibility cuts equal the original ones, i.e., $\mathcal{D}_k^{\text{corr}} = \mathcal{D}_k$.

Proof The proof follows similarly to Lemma 2. $\square$

3.4 Properties of the introduced techniques

Lemma 8 If the integer set Y is finite (Assumption 1), Algorithm 1 enhanced with gradient correction and amplification procedure for the Benders cuts, and correction of the infeasibility cuts, stops within a finite number of iterations.

Proof Infeasibility cuts (28) preserve the exclusion of previously visited infeasible points whenever a feasible incumbent exists, while ensuring that the current feasible best solution is not excluded. If a feasible solution is not found, it might happen that an infeasibility cut makes the search space an empty set. Therefore, $\text{LB} = +\infty$ and Algorithm 1 terminates. If a feasible solution is found, the modified Benders cuts $J(y_i) + \rho \tilde{g}^\top (y - y_i) \leq J(y_{b(k)})$, for all $i \in \mathbb{T}_k$, guarantee that $y_{b(k)}$ is not excluded from the search space. Algorithm 1 follows the same flow as described in the first part of the proof of Theorem 1. Therefore, we focus on the final case. Specifically, the solution of $\mathcal{P}_{\text{LB-MILP}}$ has two possible outcomes: a new $\tilde{y} \neq y_k$ or $\tilde{y} \equiv y_{b(k)}$. For the former case Algorithm 1 proceeds to the next iteration. For the latter case the algorithm directly terminates since $\text{LB} = \text{UB}$. $\square$

Theorem 2 Under Assumptions 1, 2, and 3, Algorithm 1 enhanced with the gradient correction and amplification procedure for the Benders cuts, and correction of the infeasibility cuts, terminates in a finite number of iterations and returns either a globally optimal solution of $\mathcal{P}_{\text{MINLP}}$ or a certificate of infeasibility.

Proof Assumption 3 makes unnecessary the use of gradient correction and amplification for the Benders cuts. This follows from Lemma 6 and Remark 4. Also, the correction of infeasibility cuts is not required as shown in Lemma 7. Therefore, Algorithm 1 is unchanged. Its convergence follows from Theorem 1. $\square$

Finally, we aim to present an additional result for a class of MINLPs larger than the convex one.

Assumption 4 We assume that the set $\bar{\mathcal{F}}$ is convex.

Assumption 4 includes MINLPs that have nonconvexities in the space of continuous variables but are convex in the space of integer variables. A practical example of such a case is a nonlinear control system where the discrete controls enter affinely in the dynamics, cf. the system dynamics (35) of the example in Sect. 4.2, and real-world applications in [16, 54]. We can show that Lemma 2 and Lemma 7 also hold in this case since the cuts are constructed in the integer space only.

Lemma 9 *Under Assumptions 1, 2, and 4, if $\mathcal{P}_{\text{NLP}}$ is infeasible for $\hat{y}$, and if $\bar{y}$ solves $\mathcal{P}_{\text{FNLP}}(\hat{y}, y_b)$, then inequality $(\hat{y} - \bar{y})^\top (y - \bar{y}) \leq 0$ is valid for all $y \in \bar{\mathcal{F}}$.*

Proof The proof follows similarly to Lemma 2. $\square$

Theorem 3 *If Assumptions 1, 2, and 4 hold, then Algorithm 1 enhanced with gradient correction and amplification procedure for the Benders cuts, and correction of the infeasibility cuts, terminates in a finite number of iterations and returns either a feasible solution to $\mathcal{P}_{\text{MINLP}}$ or a certificate of infeasibility.*

Proof The theorem is trivial if Algorithm 1 is initialized with a feasible solution. This follows from Lemma 8. In the other case, $b(k) \notin \mathbb{T}_k$, the constraint set for $\mathcal{P}_{\text{BR-MIQP}}$ is given by

$$\mathbb{A}_k = \{y \in Y \mid (y_i - \bar{y}_i)^\top (y - \bar{y}_i) \leq 0, \forall i \in \{0, \dots, k\}\}.$$

Lemma 9 holds for each $(y_i, \bar{y}_i)$. Therefore, $\mathcal{F} \subseteq \mathbb{A}_k$. Every visited infeasible point y_i is excluded from $\mathbb{A}_k$. Therefore, Algorithm 1 either terminates with a certificate of infeasibility since $\mathcal{F} = \emptyset$ or with a feasible point $y_k \in \mathcal{F}$. $\square$

3.5 Final details and options of algorithm 1

Algorithm 1 with early termination heuristic. We introduce a heuristic for Algorithm 1 which has shown to be particularly effective in finding high-quality solutions in a short time. The heuristic consists of a different stopping criterion where the lower bound LB is set equal to V_{MIQP} , the objective of the MIQP, and it is given by

$$\text{UB} \geq \text{LB} := V_{\text{MIQP}}. \quad (30)$$

When this heuristic is selected, Algorithm 1 does not solve any $\mathcal{P}_{\text{LB-MILP}}$ and thus no valid lower bound on the MINLP solution is computed. This algorithm is available in the CAMINO software package² with the name `s-b-miqp-early-exit`, and is denoted in the following as S-B-MIQP-ee.

Initial guess y_0 . We obtain the initial guess for the integer variables $y_0 \in Y$ by solving $\mathcal{P}_{\text{BR-MIQP}}$, which is constructed by linearizing around the solution of the relaxed $\mathcal{P}_{\text{NLP}}$ computed in line 2 of Algorithm 1. An issue may arise if the constructed $\mathcal{P}_{\text{BR-MIQP}}$

² <https://github.com/minlp-toolbox/CAMINO>.

is infeasible, however it has never happened in our computations. If this happens, it is possible to compute an initial guess using one of the heuristic methods available in CAMINO, e.g., the feasibility pump [8]. CAMINO allows concatenating solver calls so that the solution computed by the first solver is used as the initial guess for the next one. For example, to call the feasibility pump (fp) and then use its solution as the initial guess for S-B-MIQP, it is sufficient to specify the solver `fp+s-b-miqp` in CAMINO.

Exact Hessian in $\mathcal{P}_{\text{BR-MIQP}}$ for nonconvex MINLPs. When solving nonconvex MINLPs, the exact Hessian of the Lagrangian of $\mathcal{P}_{\text{MINLP}}$ constructed using primal and dual information of the incumbent solution may become indefinite. However, we want the Hessian $B_{b(k)}$ used in $\mathcal{P}_{\text{BR-MIQP}}$ to be positive semidefinite such that we always solve convex MIQPs. In the code implementation, we perform an eigenvalue decomposition of the exact Hessian. In case negative eigenvalues are detected, we perform a simple Hessian regularization by adding a diagonal matrix weighted by the absolute value of the smallest eigenvalue, ensuring positive semidefiniteness. Furthermore, if the smallest eigenvalue is below a threshold (e.g., 10^{-8}), we neglect the Hessian and revert to a linear objective.

Solution pool. An important option of MILP/MIQP solvers that we utilize is the *solution pool*, which corresponds to a set of feasible solutions found during the solution process. The user defines the maximum dimension of the solution pool N_{sp} , i.e., how many integer feasible solutions it should contain. Usually, the MIP solver operates in a best-effort manner, storing in the solution pool the best integer solution found during the branch-and-cut procedure. Thus, if the solution of the MIP is successful the solution pool contain at least one element and at most N_{sp} . Naturally, we have to solve the corresponding $\mathcal{P}_{\text{NLP}}$ for each solution stored in the pool. By means of the *solution pool*, we obtain multiple cuts per MIP solved, making Algorithm 1 more efficient.

Additional outer approximation cuts. The implementation of Algorithm 1 within CAMINO includes an option to add standard outer approximation cuts to the master problems [28]. Adding these cuts reduces the computation time in the conducted experiments. Specifically, given an integer solution y_k , the implementation adds the outer approximation cut to lower bound the objective function only when $\mathcal{P}_{\text{NLP}}$ is feasible, and the cuts correspond to

$$f(x_k, y_k) + \nabla f(x_k, y_k)^\top \begin{pmatrix} x - x_k \\ y - y_k \end{pmatrix} \leq 0, \quad k \in \mathbb{T}_k. \quad (31)$$

Moreover, the gradient of (31) is corrected and amplified in case the best solution has a better objective, i.e., $J(y_{b(k)}) \leq J(y_k) = f(x_k, y_k)$. The correction and amplification procedure is similar to the one described in Sect. 3.1-3.2. Regarding the outer approximation cuts that overapproximate the feasible set, the implementation only adds cuts corresponding to convex constraints. In the current implementation, there is a mechanism that automatically detects linear constraints, but generic convex constraints should be labeled by the user when the MINLP is formulated. In each iteration $k \in \mathbb{T}_k$, the outer approximation cuts are defined as

$$g_L^i(x, y; x_k, y_k) \leq 0, \quad i \in \mathbb{C}_g \subseteq \mathbb{Z}_{[1, n_g]}, \quad (32)$$

where $\mathbb{C}_g$ contains the indices of inequality constraints where $g^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ is convex in x, y jointly.

4 Numerical results

In this section, we illustrate the performance of Algorithm 1 via numerical experiments. First, we compare the proposed algorithm against three open-source solvers, Bonmin [11], SCIP [3, 9], and SHOT [45, 46], and the commercial solver Gurobi [34], on many MINLPs selected from the MINLPLib [19, 49]. SCIP and Gurobi are global solvers for nonconvex MINLPs and they implement a spatial branch-and-bound method. Secondly, we consider two optimal control problems (OCPs) for a nonlinear system with binary control input: a textbook example from [53, §8.17] and a real-world case study of a building climate system from [16]. Since SCIP, SHOT and Gurobi for MINLPs do not have a direct interface with CasADi, comparing it on these OCPs is nontrivial. For the simpler textbook example, we reformulated the problem in Pyomo [20, 35] to enable calling SCIP, SHOT and Gurobi. We did not attempt the same for the more complex climate-system case, because CasADi cannot directly output an AMPL description, and reformulating the problem in Pyomo is not trivial. All the presented results are obtained on a computer with an Intel(R) Xeon(R) W-2225 CPU @ 4.10GHz processor with 4 cores and 32 GB of memory. Since the benchmark on MINLPLib instances takes several hours, we ran the solvers in parallel, in batch of three, to have one core always free preventing overloading. To minimize oscillations in compute power, we disabled CPU boost and set the maximum CPU clock to 4GHz, slightly lower than the nominal CPU clock.

4.1 MINLPLib instances

This section compares Algorithm 1 (S-B-MIQP) and its variant with early termination (S-B-MIQP-ee) with four existing solvers, SHOT v1.1 [45, 46], Bonmin v1.8 [11], SCIP v9.2.2 [3, 9], Gurobi v13.0.0 [34] on a subset of instances from MINLPLib [19, 49]. The problem instances are selected according to the following requirements: 1) mixed-integer and mixed-boolean variables with nonlinear constraints and/or objective, 2) the instances have a nl-file representation compatible with CasADi nl-reader. Based on these criteria, 233 convex MINLP instances and 263 nonconvex MINLP instances were selected. Point 2) excludes instances using AMPL suffixes and some other special syntax, only 7 instances are excluded by this criterion. S-B-MIQP, S-B-MIQP-ee, Bonmin are available in the CAMINO software package³ with the name `s-b-miqp`, `s-b-miqp-early-termination` and `bonmin`, respectively. SHOT is called from command-line interface, loads the MINLPLib instance as a `.nl` file and solves it. SCIP and Gurobi (for MINLPs) are called using the AMPL Python interface which loads the MINLPLib as `.mod` file. For reproducibility, the results presented in this section are available in the public repository CAMINO-benchmark⁴.

³ <https://github.com/minlp-toolbox/CAMINO>.

⁴ <https://github.com/minlp-toolbox/CAMINO-benchmark>.

S-B-MIQP, Bonmin and SHOT are configured to use the same NLP-solver, Ipopt 3.14 [61] with ma27 [36] to have a fair comparison. The MIP subproblems are solved using Gurobi 13.0.1 [34] on a single thread with a solution pool of dimension 10 for SHOT and 5 for both S-B-MIQP and S-B-MIQP-ee. SCIP and Gurobi utilize their algorithms for MINLPs. For each problem, the maximum wall time is 300 seconds. When the time limit is reached, we return the feasible solution with the best objective if available, otherwise we consider it as a failure. The primal solution satisfies a tolerance of 10^{-8} on both the objective and constraints. The MINLP gap for termination is set to 10^{-2} . The nl-file of each problem contains an initial guess. For both S-B-MIQP algorithms, the initial point $y_0 \in Y$ is generated using the strategy described in Sect. 3.5. Specifically, we take the initial point provided in the nl-file and use it as a warm start for IPOPT to solve the integer relaxation of $\mathcal{P}_{\text{MINLP}}$. The solution of this relaxation is then used to construct the first $\mathcal{P}_{\text{BR-MIQP}}$, whose solution yields y_0 . The parameters of both S-B-MIQP algorithms are set as follows: the hyperparameter $\alpha = 0.5$, the gradient amplification $\rho = 1.5$, the weight matrix W is the identity matrix, and B is the exact Hessian in $\mathcal{P}_{\text{BR-MIQP}}$. The exact Hessian is constructed by evaluating the second-order derivatives of the cost and nonlinear constraints of $\mathcal{P}_{\text{MINLP}}$ at the best solution found, which also include the optimal multipliers corresponding to the nonlinear constraints. At the end of this subsection we present a sensitivity analysis for tuning α and ρ .

We compare the algorithms using performance profiles for the objective value of the solutions and for the wall time. The wall time is computed in the same way for each algorithm, by starting a clock when the respective algorithm is called and by stopping the clock when the termination condition is met. For the performance profile of the wall time we used the standard method described in [24]. Consider $q_{p,s}$ the quantity of interest required to solve problem $p \in \mathcal{P}$ by solver $s \in \mathcal{S}$, where $\mathcal{P}$ is the set of problems and $\mathcal{S}$ is the set of solvers. Then, the performance ratio is defined as

$$r_{p,s} = \frac{q_{p,s}}{\min\{q_{p,s} \mid s \in \mathcal{S}\}}, \quad (33)$$

in case solver s cannot solve problem p , the ratio $r_{p,s} = \infty$. The performance profile is defined as

$$\pi_s(\tau) = \frac{\text{card}(p \in \mathcal{P} \mid r_{p,s} \leq \tau)}{\text{card}(\mathcal{P})}, \quad (34)$$

where card denotes the cardinality of a set. Thus, $\pi_s(\tau)$ is the probability for solver $s \in \mathcal{S}$ that a performance $r_{p,s}$ is within factor $\tau > 0$ of the best possible ratio. The function π_s is the cumulative distribution function of the performance ratio.

Since the objective value of a problem $p \in \mathcal{P}$ can be negative, in order to compute the performance profile we apply a scaling to obtain only positive numbers,

$$\text{if } q^* = \min\{q_{p,s} \mid s \in \mathcal{S}\} < 0 \text{ then } \hat{q}_{p,s} = q_{p,s} - q^* + 1, \quad s \in \mathcal{S}.$$

For the 233 convex MINLPs, we summarize the results in Table 2. We define as “success” the instances that reached the desired MINLP gap within time limit, as “failures” the instances where a feasible solution is not available at the time limit,

Table 2 Summary of benchmark for convex MINLPs

Solver	Success	Fail	Time-out	Gap $< 10^{-1}$
Bonmin	128	66	39	27
Gurobi	202	2	29	23
SCIP	185	2	46	38
SHOT	212	9	12	10
S-B-MIQP	200	7	26	19
S-B-MIQP-ee	224	8	1	1

and as “time-out” the instances where a feasible solution is available, but the desired gap is not yet achieved. For the latter case, we report the number of instances whose gap is smaller than 0.1, an interval 10 times larger than the sought MINLP gap. We report S-B-MIQP-ee at the bottom of Table 2 since it is a heuristic method even for convex MINLPs, so the status “success” does not necessarily correspond to the global minimizer found rather to a feasible solution found.

Figure 9 depicts the performance profiles of objective value and wall time achieved by the different algorithms on the set of convex MINLPs. Regarding wall time, Gurobi and SHOT are the fastest solvers that guarantee to find globally optimal solutions. S-B-MIQP is about 4 times slower than SHOT. S-B-MIQP and SCIP have similar wall-time profiles, but we can claim that S-B-MIQP is slightly better since it achieves 202 successes within the available time while SCIP achieves 185 successes. Nevertheless, SCIP is a bit more robust as it fails only for 2 instances (*fac2* and *tls12*) while S-B-MIQP fails for 7 instances (*ibs2*, *o7*, *p_ball_30b_10p_2d_h*, *tls12*, *tls5*, *tls6*, *tls7*). SHOT fails for 9 instances, namely *ibs2* and the *p_ball* problems *10b_7p_3d_h*, *20b_5p_2d_h*, *30b_10p_2d_h*, *30b_5p_2d_h*, *30b_5p_3d_h*, *30b_7p_2d_h*, *40b_5p_3d_h*, and *40b_5p_4d_h*. Gurobi fails for the *p_ball* instance *40b_5p_4d_h* and *tls12*. Bonmin solves fewer problems overall and needs more time to do so compared to the other algorithms. The heuristic algorithm S-B-MIQP-ee is of course faster than S-B-MIQP as it does not solve any $\mathcal{P}_{\text{LB-MILP}}$, and on many instances is the second- or third-fastest algorithm. Moreover, S-B-MIQP-ee finds globally optimal solutions for about 70% of the problems (cf. left plot of Fig. 9). However, this cannot be proven in general. Overall SHOT is the fastest open-source algorithm; yet, in the next subsection we show that, once the Python overhead is removed, S-B-MIQP can achieve similar performance.

We comment on the instances where S-B-MIQP fails. Instance *ibs2* is a relatively large-size problem with few nonlinear constraints having a dense Jacobian, we believe that there is some issue with its *nl*-file representation, as it fails for all solvers reading *nl*-files, while it solved successfully using SCIP and Gurobi which read *mod*-files. For *o7* the solution of $\mathcal{P}_{\text{BR-MIQP}}$ is quite slow as the MIQP solver struggles to improve lower bounds. By implementing a watchdog on the gap of the MIQP solver, we could abort the computation and return the best feasible solution on the tree, without waiting that the tolerance on the gap of $\mathcal{P}_{\text{BR-MIQP}}$ is met. In this way S-B-MIQP can proceed, generate more cuts and find the solution. We tested this watchdog strategy and successfully solved *o7*. However, the strategy is not included in the main release of *s-b-miqp*.

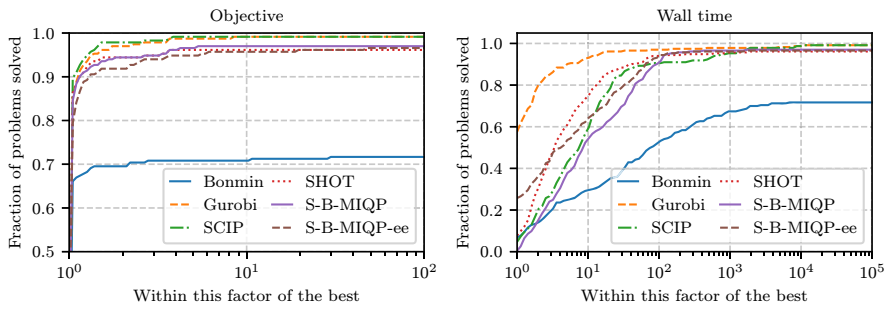


Fig. 9 Convex MINLPs. Comparison of Algorithm 1 with standard termination (S-B-MIQP) and with early termination (S-B-MIQP-ee), cf. Sect. 3.5, against Bonmin, Gurobi, SCIP, and SHOT. Left: Performance profiles for the objective value achieved by each solver. Right: Performance profiles for the wall time of each solver

For the instance `p_ball_30b_10p_2d_h`, S-B-MIQP performs six iterations in the given time budget but it is unable to compute a feasible solution. The problem might be particularly hard since MINLPLib does not report a valid dual bound. For instances `tls`, S-B-MIQP is stuck in a loop that solves feasibility NLP and adds feasibility cuts to $\mathcal{P}_{\text{BR-MIQP}}$. The infeasibility cuts seem to work properly, since the objective of $\mathcal{P}_{\text{NLP}}$ is decreasing during the iterations. We suspect numerical issues in the solution of $\mathcal{P}_{\text{NLP}}$, also because once the integer variables are fixed, $\mathcal{P}_{\text{NLP}}$ is overconstrained, having more equality constraints than optimization variables. We believe that all these failures could be mitigated by implementing some reformulation routines and bound tightening for the MINLP, as done for instance in SHOT. Table 7 in the Appendix reports the objective and wall time achieved by each algorithm.

We summarize the results for nonconvex MINLPs in Table 3. Differently from above, all algorithms except SCIP and Gurobi are only heuristics for nonconvex MINLPs. Therefore, we classify as “success” a feasible solution produced by the solver when its termination is triggered. We classify as “fail” any instance where the solver cannot find a feasible solution within the time limit. Finally, the instances counted as “time-out” correspond to the best feasible solution found by the solver when the time limit is reached. Since SCIP and Gurobi are global solvers, comparing their solution time against the other solvers would not be meaningful as we expect that the spatial branch-and-bound procedure they execute to compute valid lower bounds is much more expensive than simply finding feasible solutions. Therefore, on each instance we limit the available time of SCIP and Gurobi to the computation time achieved by S-B-MIQP. In case S-B-MIQP failed for a specific instance, we consider the standard time limit of 300 seconds. Doing so we obtain a comparison among primal heuristics. We remark that the termination conditions of S-B-MIQP, SHOT, and Bonmin are unchanged, however, for nonconvex MINLPs the computed lower bounds might be wrong. Thus, the corresponding termination conditions might be triggered for feasible solutions that are not global minimizers.

For nonconvex MINLPs, Gurobi solves the largest share of problems, closely followed by S-B-MIQP and SCIP. Bonmin and SHOT close the ranking with a percentage of failures close to 50%. Fig. 10 depicts the performance profiles of objective value

Table 3 Summary of benchmark for nonconvex MINLPs. For SCIP and Gurobi “success” correspond to a globally optimal solution, for the other solvers not. For SCIP and Gurobi the time limit is set equal to the computation time achieved by S-B-MIQP on the same instance

Solver	Success	Fail	Time-out	Gap $< 10^{-1}$
Bonmin	133	121	15	-
SHOT	128	133	8	-
S-B-MIQP	177	82	10	-
S-B-MIQP-ee	176	89	4	-
Gurobi	170	73	26	21
SCIP	155	85	29	22

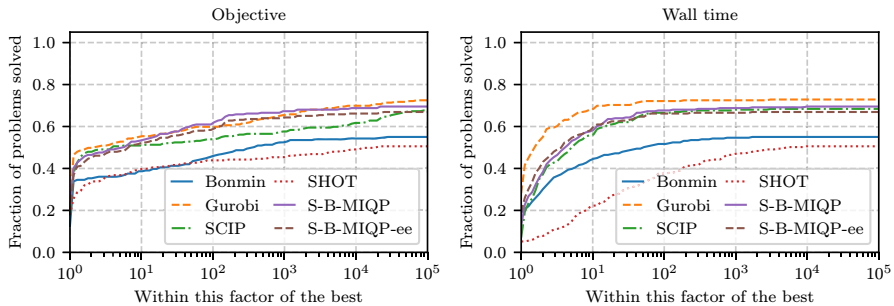


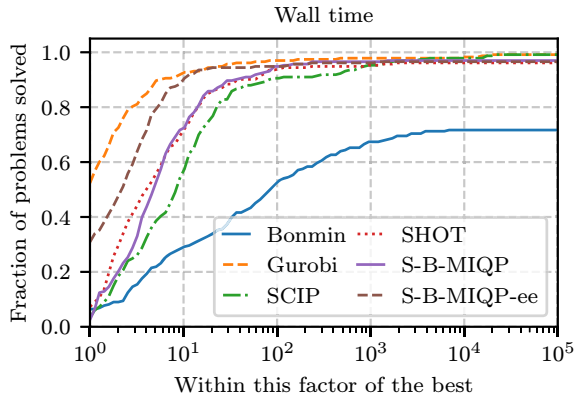
Fig. 10 Nonconvex MINLPs. Comparison of Algorithm 1 with standard termination (S-B-MIQP) and with early termination (S-B-MIQP-ee), cf. Sect. 3.5, against Bonmin, Gurobi, SCIP, and SHOT. Left: Performance profiles for the objective value achieved by each solver. Right: Performance profiles for the wall time of each solver

and wall time achieved by the different algorithms on the set of nonconvex MINLPs. Regarding the objective value, S-B-MIQP, Gurobi, and SCIP find the same solutions for about 40% of the problems. For the remaining 30%, SCIP is slightly dominated. Bonmin and SHOT perform well only for 25% of the problems, on the remaining problems they yield worse solutions than the other solvers. Regarding the wall time, Gurobi is the fastest solver. Despite having set its time limit identical to the one of S-B-MIQP, for some instances where S-B-MIQP reaches the time limit Gurobi converges to the global minimizer in less time. Similarly, in some cases Gurobi can find feasible solutions while the other solver fails. Conversely, the wall-time profile of SCIP is very much overlapped with the one of S-B-MIQP as they share the same time limit. S-B-MIQP-ee is only marginally better than S-B-MIQP, showing that for the majority of the problem, when $\mathcal{P}_{LB-MILP}$ is solved, it returns a solution with $V_{MILP} \geq UB$, triggering immediate termination of S-B-MIQP. The wall-time profiles of Bonmin and SHOT are completely dominated by the other solvers. SHOT is particularly slow, suggesting that its algorithm might not be a good heuristic for nonconvex MINLPs. Table 8 in the Appendix reports the objective and wall time achieved by each algorithm.

4.1.1 Considering only solver time for S-B-MIQP algorithms

To illustrate the algorithmic potential of S-B-MIQP for solving convex MINLPs, we evaluate the solver time independently of the Python framework overhead. Recall that

Fig. 11 Performance profile for wall time obtained on convex MINLPs. Differently from the right plot of Fig. 9, for S-B-MIQP and S-B-MIQP-ee we considered as part of the wall time only the time spent into the subsolvers, disregarding the time spent in Python operations. This figure evidences that the algorithmic logic of S-B-MIQP competes with SHOT, suggesting that an implementation in compiled code would be state-of-the-art



CAMINO implements S-B-MIQP using Python for high-level operations, while lower-level solver interfacing relies on CasADi. In contrast, solvers like SHOT or SCIP are implemented in compiled code and do not incur this Python overhead. Therefore, in Fig. 11, we present a performance profile considering only the wall time spent solving the subproblems; this excludes operations performed in Python, such as cut generation and management. Comparing Fig. 11 with the right plot in Fig. 9, we observe that S-B-MIQP performs comparably to SHOT and outperforms SCIP. Furthermore, the S-B-MIQP-ee heuristic emerges as the second-fastest solver. These results demonstrate that a compiled implementation of S-B-MIQP would achieve state-of-the-art performance. Crucially, the algorithm currently operates without the advanced presolving routines standard in SHOT, such as bound tightening, model reformulation, and constraint disaggregation, suggesting that its performance could even improve.

4.1.2 Sensitivity analysis for tuning α and ρ

Since the hyperparameter α used in (12) enters only the computation of the Benders region in $\mathcal{P}_{\text{BR-MIQP}}$, we tune its value using the algorithm S-B-MIQP-ee. If we would use S-B-MIQP, the benefit of choosing different α is less evident since the overall algorithm wall-time is influenced also by the solution of $\mathcal{P}_{\text{LB-MILP}}$. We compared S-B-MIQP-ee with $\alpha \in \{0.05, 0.25, 0.5, 0.75, 0.95\}$ on convex MINLP instances, reporting performance profiles for both objective value of the solution and wall time. As shown in the left plot of Fig. 12, S-B-MIQP-ee is largely insensitive to α with respect to the objective value, with differences emerging on only about 4% of the problems. Conversely, the choice of α has a strong impact on wall time (see the right plot of Fig. 12), where lower values of α dominate the performance profile. These results align with our expectations: an α close to zero produces a smaller Benders region, forcing $\mathcal{P}_{\text{BR-MIQP}}$ to find a solution that is lower bounded by all Benders cuts while having an objective smaller than the reduced right-hand side, cf. (12). With small α , it is more likely than the feasible set of $\mathcal{P}_{\text{BR-MIQP}}$ becomes empty, thus triggering the termination of S-B-MIQP-ee. We selected $\alpha = 0.5$ as the default value for S-B-MIQP algorithms. Although $\alpha = 0.05$ and $\alpha = 0.25$ are marginally faster on average, they failed to solve the `gams01` instance within the time limit.

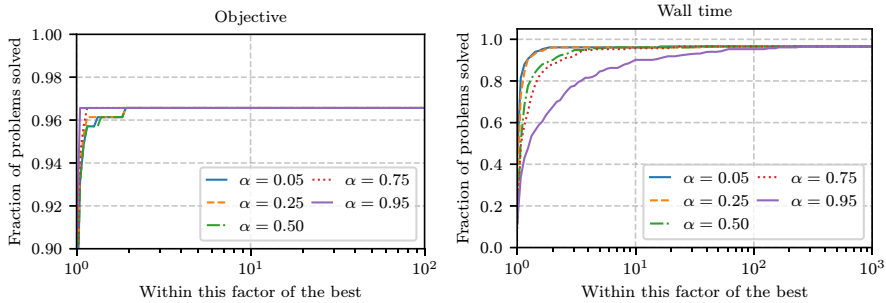


Fig. 12 Sensitivity analysis of S-B-MIQP-ee to the hyperparameter α used to shrink the Benders region of $\mathcal{P}_{\text{BR-MIQP}}$ according to (12). Note the different scales of the axes. The considered problems are the 233 convex MINLPs instances selected from MINLPLib

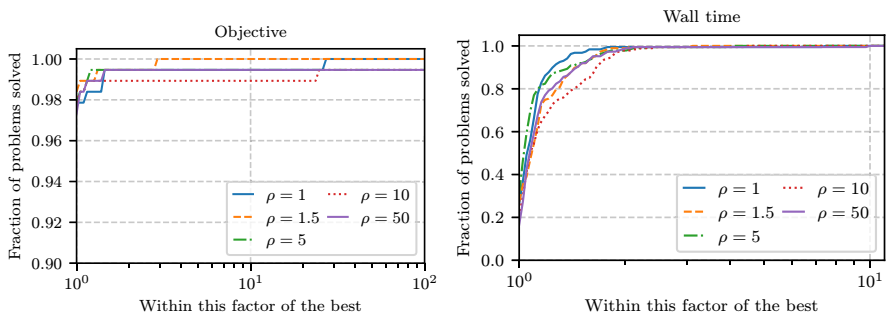


Fig. 13 Sensitivity analysis of S-B-MIQP to the hyperparameter ρ used to amplify the gradient of the corrected cuts according to (27). Note the different scales of the axes. The considered problems in the analysis are 187 instances among the nonconvex MINLPs where the cut correction routine (22) is called

To tune the hyperparameter ρ used to amplify the gradient of the corrected cuts according to (27), we focused on the 187 nonconvex instances where the cut correction routine of S-B-MIQP was invoked. Figure 13 depicts the performance profile for objective value and wall time obtained by running S-B-MIQP with $\rho \in \{1, 1.5, 5, 10, 50\}$. In terms of objective value, the algorithm converges to the same solution for approximately 98% of the 187 problems. The variant with $\rho = 1.5$ achieves slightly better performance. Notably, the instance *mbtd* reaches the time limit for all $\rho > 1.5$. Concerning wall time, the plot is dominated by $\rho = 1$, which is expected, since without gradient amplification S-B-MIQP terminates as soon as a feasible solution is found. Overall, the values $\rho \in \{1.5, 5, 50\}$ perform very similarly, while $\rho = 10$ results in slightly longer wall times. We selected $\rho = 1.5$ as the default value for S-B-MIQP algorithms, as it achieves the best solution quality while remaining competitive in computation time.

4.2 MIOC of an unstable nonlinear system

Consider a reference tracking task for a nonlinear unstable system with discrete control input taken from [53, §8.17]. The dynamic is described by the ordinary differential

equation (ODE)

$$\dot{x}(t) = x^3(t) - u(t), \quad t \in [t_0, t_f] \quad (35)$$

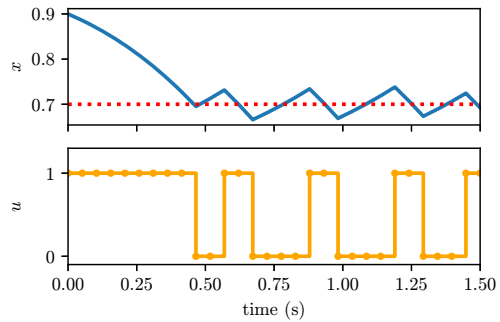
where the state is denoted by $x(t) \in \mathbb{R}$ and the control by $u(t) \in \{0, 1\}$. The control is subject to a minimum dwell time constraint of 0.1 s. The aim is to track the state reference $x_{\text{ref}} = 0.7$, starting from the initial state $\bar{x}_0 = 0.9$. By means of the multiple shooting approach for direct optimal control [10], we discretize the ODE over a fixed grid with $N = 30$ intervals such that $t_0 < t_1 < \dots < t_N = t_f$ adopting a 4th order explicit Runge-Kutta (RK) integrator and a sampling time $t_s = 0.05$ s. The resulting discretized OCP is

$$\begin{aligned} \min_{\substack{x_0, u_0, \dots, \\ u_{N-1}, x_N}} \quad & \sum_{k=0}^N (x_k - x_{\text{ref}})^2 \\ \text{s.t.} \quad & x_0 = \bar{x}_0, \\ & x_{k+1} = F_{\text{RK}}(x_k, u_k), \quad k = 0, \dots, N-1, \\ & u_k \in \mathcal{U} \quad k = 0, \dots, N-1, \end{aligned} \quad (36)$$

where $\mathcal{U} := \{u \in \{0, 1\}^{N-1} \mid u_k \geq u_{k-1} - u_{k-2}, \quad k = 0, \dots, N-1\}$ imposes a minimum uptime for the control inputs of two consecutive time steps. The required previous values u_{-1}, u_{-2} are set to zero. Moreover, x_{ref} denotes the state reference to track and function F_{RK} corresponds to the RK integrator. Problem (36) is a nonconvex MINLP which we solved using different algorithms. The results are listed in Table 4, and Fig. 14 depicts the globally optimal state and control trajectories of (36). In Table 4 we divided the solvers into two sets. In fact, for Bonmin, CIA [55] and S-B-MIQP with Gauss-Newton (GN) Hessian approximation we formulated the problem directly in CAMINO using CasADi. The native formulation in CAMINO allows us to have more control over the algorithms, for instance, we specified the Hessian approximation to use in $\mathcal{P}_{\text{BR-MIQP}}$ of S-B-MIQP. Also, we can flag dwell-time constraints such that they are dropped from $\mathcal{P}_{\text{NLP}}$ of S-B-MIQP. Doing this improves the solution of the relaxed $\mathcal{P}_{\text{MINLP}}$ used to compute the starting point $y_0 \in Y$. The other solvers, Gurobi, SCIP, and SHOT, are called via AMPL and they load an nl-file representation of (36) generated via Pyomo, the model is open-source⁵. We also solve the nl-file representation of (36) via S-B-MIQP to show the performance of the proposed algorithm when it is used as a “black box”. Indeed, the nl-file does not allow us to specify the same additional information in the model formulation. Therefore, in this case the exact Hessian is used in $\mathcal{P}_{\text{BR-MIQP}}$ and dwell-time constraints will be part of each $\mathcal{P}_{\text{NLP}}$.

The MINLP gap is set to 10^{-4} , solvers share the same parameters when possible and multithreading is allowed. Bonmin ran with its default nonlinear branch-and-bound routine. S-B-MIQP utilized Gurobi as solver for its MILP/MIQP master problems with solution pool dimension equal 5. The CIA master problem was solved with the tailored branch-and-bound solver pycombina [15]. Finally, the two global solvers, SCIP and Gurobi ran their default spatial branch-and-bound algorithm. For further implementation details, we refer the reader to the collection of problems in the CAMINO

⁵ https://github.com/minlp-toolbox/misc/blob/main/pyomo_unstable_ocp.py.

Fig. 14 Optimal state and control trajectories of (36)**Table 4** Comparison of different algorithms for the solution of (36)

Algorithm	Objective	Runtime [s]
<i>Natively in CAMINO</i>		
Bonmin	0.1765	11.43
CIA [16]	0.1771	0.029
S-B-MIQP (w. GN Hessian)	0.1765	0.35
<i>Loading .nl file</i>		
Gurobi*	0.1765	6.16
SCIP*	0.1765	3.92
SHOT	0.1775	3.43
S-B-MIQP (w. exact Hessian)	0.1792	2.58

*Global solvers.

repository. We noticed that both Bonmin and S-B-MIQP with GN Hessian found the global optimum reported in [53, §8.17], while the specialized algorithm CIA returned a slightly suboptimal solution but in only 0.029 seconds. Interestingly, S-B-MIQP with GN Hessian returns the solution in only 0.35 seconds while Bonmin takes more than 10 seconds. The two global solvers SCIP and Gurobi find the global optimum within a few seconds. A similar time is required by SHOT and S-B-MIQP with exact Hessian but they found suboptimal solutions even worse than CIA.

4.3 MIOC of a renewable energy system

To assess the performance of Algorithm 1 on an example of real-world complexity, we consider an extended version of the solar-thermal-climate-system (STCS) described in [16] and physically installed at Karlsruhe University of Applied Sciences. The system provides cooling for the building's main hall by means of thermal machines driven by solar-thermal energy. Specifically, the system consists of an absorption cooling machine (ACM) and a heat pump (HP). The first machine can be used in absorption cooling (AC) mode, during which the solar thermal heat drives the machine ($b_{ac} = 1$), or in free cooling (FC) ($b_{fc} = 1$), where the cooling tower installed on the roof of the building can directly cool down the medium at ambient temperature. The second machine is a regular HP that has been installed more recently to provide flexibility

and extra cooling power. When it is switched on ($b_{hp} = 1$), it immediately provides cooling energy since the machine is connected to the low-temperature storage and driven by electric energy. The electric energy required to drive the system is either bought from the grid or supplied by on-site PV panels. A schematic of the current plant is contained in [32]. Experimental operations and numerical case studies have been carried out on this system in [13, 14, 16, 17, 32].

The system dynamics are modeled via a set of ODEs with $\xi \in \mathbb{R}^{19}$ differential states, $\mu \in \mathbb{R}^6$ continuous controls and $v \in \{0, 1\}^3$ binary controls. The system state includes the temperature of the flat plate and vacuum tube solar collectors, T_{fpsc} and T_{vtsc} respectively, four temperature levels in the stratified high-temperature storage, $T_{ht,i}$, $i = 1, 2, 3, 4$, the temperature of the low-temperature storage T_{lt} , and the temperature inside the primary and secondary solar circuit, T_{psc} and T_{ssc} , respectively. The continuous controls regulate the electric power absorbed from the grid, velocity and pressure of the pumps in the solar circuit, and the input/output flow rate of the high-temperature storage. As described above, there are three binary controls that decide the switching on/off of the different machines. The system is subject to several ambient conditions, represented as time-varying parameters in the NLP. These are the ambient temperature, the solar irradiance on the solar collectors, the power generated by the local PV panels, the price of electric energy, and the desired cooling load profile.

It is now possible to formulate a MIOCP that aims to provide the specified cooling power while operating the system safely and energy-efficiently. Using a direct approach, we discretize the MIOCP, which has a time horizon of 24 hours, via Gauss-Radau collocation of order 3. We divide the time horizon into $N = 24$ intervals with a sampling time of 1 hour. To guarantee feasibility of the MIOCP, we introduce $n_s = 24$ slack variables in each discretization interval, which are penalized linearly and quadratically in the cost function. Therefore, the STCS problem has $N \cdot (19 \cdot d + 6 + 3 + n_s) = 2160$ variables, of which $3 \cdot N = 72$ are binary.

By a stage-wise concatenation of the variables, we define the vector of continuous and binary decision variables as x and y , respectively. Thus, the resulting MINLP can be stated compactly as

$$\begin{aligned} \min_{\substack{x \in \mathbb{R}^{n_x}, \\ y \in \{0, 1\}^{n_y}}} \quad & \|f_1(x, y)\|^2 + f_2(x, y) \\ \text{s.t.} \quad & g(x, y) \leq 0, \\ & h(x, y) = 0, \end{aligned} \quad (37)$$

where the cost function is the sum of a quadratic term, aiming to minimize constraint violation and achieve smooth actuation of the mixing valves, and a nonlinear one, defined by f_2 , which incorporates the electricity cost for operating the system. The special structure of the cost function allows for positive semidefinite Hessians for the MIOQPs via the Gauss-Newton approximation [48].

The MIOCP is modeled and solved using CAMINO. We compare the solutions obtained with S-B-MIOQP, S-B-MIOQP-ee, the specialized algorithm CIA [15, 55], and the Bonmin nonlinear branch-and-bound method. The algorithms share the same parameters where applicable. The MINLP gap for termination is set to 10^{-2} and the

Table 5 Comparison of different algorithms for the solution of STCS (37)

Algorithm	Objective	Runtime (mm:ss)	Timestamp Best Solution
<i>Relaxed</i> NLP	1547.33	00:05	-
Bonmin	2409.66	30:00	02:43
CIA [16]	6370.03	00:08	-
S-B-MIQP	2288.40	30:00	01:28
S-B-MIQP-ee	2288.40	30:00	01:28

computation time limit is set to 30 minutes for each algorithm. IPOPT [61] uses HSL ma57 [36] as the internal linear solver. The MIP subproblems are solved using Gurobi with multithreading enabled and a solution pool of dimension 3. Moreover, the MIP gap of $\mathcal{P}_{\text{BR-MIQP}}$ is set to 10%, while that of $\mathcal{P}_{\text{LB-MILP}}$ is set to 5%. The time limit for each master problem in every S-B-MIQP iteration is the maximum between 600 seconds and the remaining overall MINLP time budget. If the master problems hit the time limit, we accept the available incumbent solution, even if it has a MIP gap larger than prescribed. If no solution is available, the corresponding master problem is considered a failure. In case $\mathcal{P}_{\text{LB-MILP}}$ cannot be solved, S-B-MIQP terminates and returns the best solution found. The two hyperparameters of S-B-MIQP, α (12) and ρ (27), are set to their default values of 0.5 and 1.5, respectively. For more details regarding the model and solver options, we invite the reader to consult the published code⁶.

We report the objective values of the solutions and the runtimes in Table 5.

For the CIA algorithm, we quickly obtain a solution but the corresponding objective is fairly high, corresponding to a very suboptimal operating strategy for the energy system. For the other algorithms, the termination is triggered by reaching the time limit of 30 minutes. However, in each case the best solution is found much earlier. Bonmin finds its best solution in less than 3 minutes while S-B-MIQP takes less than 2 minutes. Differently from CIA, both Bonmin and S-B-MIQP deliver solutions that result in a high-performing operation of the plant. In Fig. 15, we plot the optimal trajectories of the binary controls and of a selected subset of the state extracted from the best solution computed by S-B-MIQP. The plots show an almost optimal operation of the system, where temperature bounds and predictions of the external parameters over the horizon have been exploited by the optimizer. However, since constraints are imposed in a soft way, the solution exhibits a small violation of the upper bound of the low-temperature storage. Note that the free-cooling mode is never active, meaning that the system never dissipates heat into the environment rather storing it in the high temperature water tank.

In Table 6, we report the results of each iteration performed by S-B-MIQP. The first row shows the objective value of the relaxed $\mathcal{P}_{\text{NLP}}$ and the objective of $\mathcal{P}_{\text{BR-MIQP}}$ constructed around the relaxed $\mathcal{P}_{\text{NLP}}$ solution, as described in Sect. 3.5. Notably, for S-B-MIQP the best solution is obtained in the first iteration, by solving $J(y_1)$. We believe this occurs because the solution of the relaxed $\mathcal{P}_{\text{NLP}}$ provides an informative starting point for constructing the first $\mathcal{P}_{\text{BR-MIQP}}$, whose solution already yields the

⁶ https://github.com/minlp-toolbox/CAMINO/blob/main/camino/problems/solarsys/_init_.py.

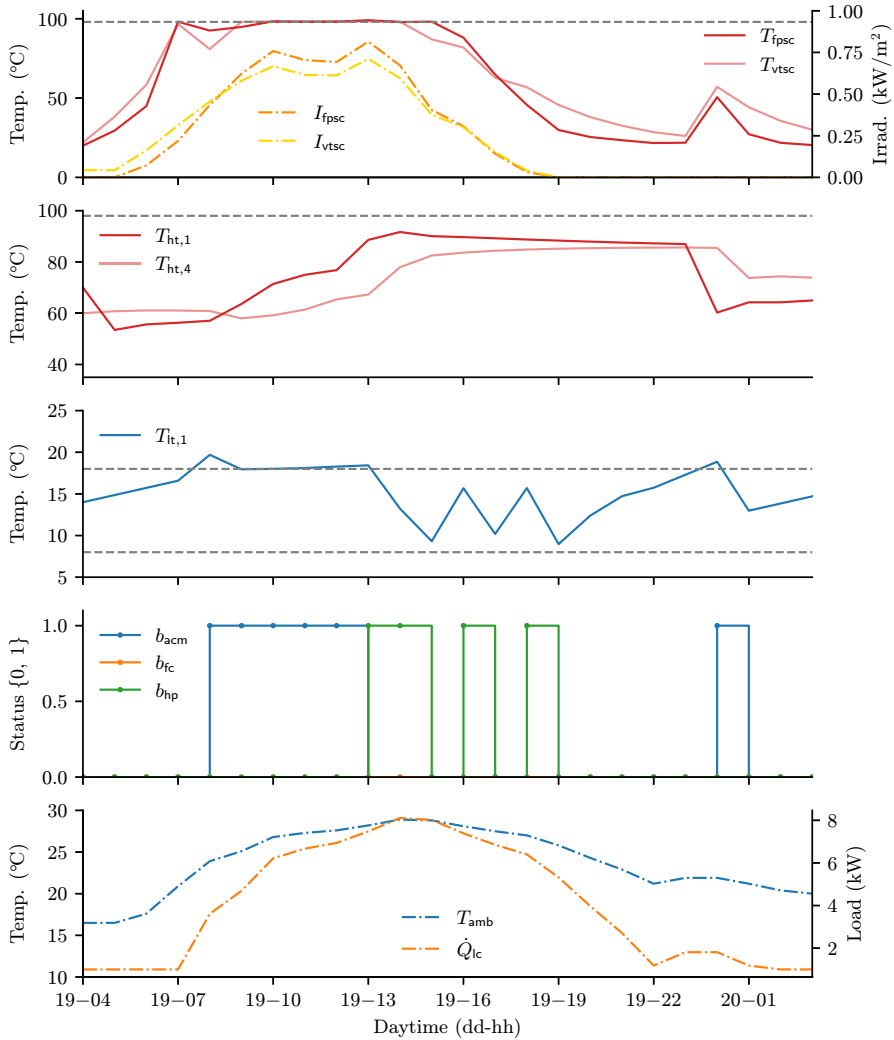


Fig. 15 State and binary control trajectory obtained by solving (37) via S-B-MIQP. The plotted solution is obtained at iteration 1 and has the lowest objective found. The bounds on state and control are the dashed gray lines. The ambient conditions are represented by dashed-dotted lines, we have solar irradiance on the flat plate and vacuum tube solar collectors, $I_{fp\text{sc}}$, $I_{vt\text{sc}}$, respectively, the ambient temperature, T_{amb} , and the cooling load profile, $\dot{Q}_{lc}$

best integer controls found. During the remaining computation time, both S-B-MIQP variants attempt to improve this solution, but without success. The last three columns of Table 6 report the wall time of each subproblem. The first two $\mathcal{P}_{\text{BR-MIQP}}$ are quite expensive to solve compared to $\mathcal{P}_{\text{NLP}}$ and the first ten $\mathcal{P}_{\text{LB-MILP}}$. From iteration 14, also the $\mathcal{P}_{\text{LB-MILP}}$ requires longer computation times. These computation times could be reduced by implementing a single-tree strategy for the solution of $\mathcal{P}_{\text{LB-MILP}}$ [2, 52]

Table 6 Iterations of S-B-MIQP for the solution of STCS (37)

Iter. k	UB	LB	$J(y_k)$	$V_{\text{MIQP},k}$	$V_{\text{MILP},k}$	NLP time	MIQP time	MILP time
0	∞	$-\infty$	1547.33*	2134.29	–	2.996	78.945	–
1	∞	1547.33	2288.40	2043.05	–	5.729	57.598	–
2	2288.40	1547.33	2438.16	–	–10475.65	4.648	–	0.819
3	2288.40	1547.33	329568.02	–	–6604.50	2.315	--	6.594
4	2288.40	1547.33	92830.58	–	–4460.51	7.080	--	0.974
5	2288.40	1547.33	94530.31	–	–4533.46	5.417	--	1.039
6	2288.40	1547.33	12753.07	–	–2511.64	4.806	–	1.180
7	2288.40	1547.33	42158.56	–	–2156.02	1.647	--	5.020
8	2288.40	1547.33	12578.73	–	–1474.22	8.012	–	1.008
9	2288.40	1547.33	18825.85	–	–1428.67	11.123	–	8.903
10	2288.40	1547.33	8360.52	–	–1412.07	6.128	–	17.219
11	2288.40	1547.33	5345.72	–	–1401.91	7.885	–	24.223
12	2288.40	1547.33	7135.18	–	–1403.42	6.328	–	50.230
13	2288.40	1547.33	4378.46	–	–1372.62	5.632	–	52.926
14	2288.40	1547.33	3890.60	–	–1348.06	6.702	–	99.330
15	2288.40	1547.33	3898.51	–	–1308.88	5.095	–	29.526
16	2288.40	1547.33	3956.89	–	–1233.70	9.644	–	123.145
17	2288.40	1547.33	2806.25	–	–1143.33	5.575	–	46.358
18	2288.40	1547.33	2735.66	–	–1112.57	7.614	–	57.206
19	2288.40	1547.33	3518.17	–	–817.74	4.942	–	47.172
20	2288.40	1547.33	2763.79	–	–645.47	7.038	–	66.940
21	2288.40	1547.33	3391.25	–	–502.84	7.150	–	104.462
22	2288.40	1547.33	2658.91	–	–266.93	5.481	–	119.463
23	2288.40	1547.33	2571.64	–	660.16	4.431	–	105.601
24	2288.40	1547.33	14361.66	–	660.25	7.390	–	130.488
25	2288.40	1547.33	11681.92	–	668.18	11.496	–	135.643
26	2288.40	1547.33	16616.70	–	670.43	9.564	–	82.265
27	2288.40	1547.33	14503.88	–	681.85	6.446	–	82.006
28 [†]	2288.40	1547.33	14557.06	–	709.98	5.919	–	79.970

*relaxed solution; [†]iterations stopped due to time limit

while adding new constraints via “lazy constraints” callbacks, available for instance in Gurobi.

To conclude, with this example we aimed to show that S-B-MIQP can be applied out-of-the-box to a large and complex MINLP with satisfactory results. In this case, the termination of S-B-MIQP was triggered by the time limit of 30 minutes but already in less than 2 minutes an integer feasible solution was available, which for this case coincided with the best solution found.

5 Conclusion and outlook

We presented a novel algorithm for solving MINLPs. The algorithm combines cutting planes based on generalized Benders' decomposition and outer approximation efficiently and converges to the global optimum or with a certificate of infeasibility for convex MINLPs. We proposed an extension for treating nonconvex MINLPs employing a heuristic to modify the generated cutting planes. The extension does not alter the results for convex MINLPs while it makes the algorithm directly applicable to nonconvex problems. The algorithm was compared to Bonmin, Gurobi, SCIP and SHOT for a large subset of both convex and nonconvex MINLPs from the MINLPLib. The results show that the proposed algorithm is particularly suited for nonconvex MINLPs while it closely follows the performance of SHOT for convex MINLPs. Finally, we presented the results obtained with the proposed algorithm in two cases of optimal control of systems with discrete actuators. The open-source software package CAMINO⁷, developed to implement the proposed algorithm, is coded in Python and relies on CasADi for modelling the optimization problems and interfacing with required solvers. Moreover, CAMINO includes implementations of various methods found in the literature. A welcome addition is an improved AMPL-file generation from CasADi for OCPs, enabling a quick interface with any AMPL-compatible solver. For higher efficiency, a direct interface between CasADi's expression graph and those of SHOT, SCIP, and Gurobi could be developed, though this would require more implementation effort.

Overall, the proposed algorithm shows promising solution quality and computation time results. This performance might be further improved by an efficient implementation in compiled code, adding preprocessing routines at the MINLP level, such as bound tightening, and algorithmic strategies like the single-tree mode used in SHOT, where the solution of the master problem can resume from previous iteration without rebuilding the branch-and-bound tree. In the future we aim at extending the proposed algorithm to become a global solver for *nonconvex* MINLPs, e.g., by using ideas from [44].

Appendix: tabular results of section 4.1

In the tables below, wall time is reported in seconds, and we use the acronym “NaN” when the algorithm fails to find a feasible solution within the 300-second time limit.

⁷ <https://github.com/minlp-toolbox/CAMINO>.

Table 7 Benchmark results for convex MINLPs

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
batch	2.86e+05	0.293	2.86e+05	0.0979	2.86e+05	0.445	2.86e+05	0.111	3.01e+05	0.114	3.09e+05	0.0995
batch0812	2.69e+06	1.78	2.7e+06	0.055	2.69e+06	0.269	2.71e+06	0.482	3.11e+06	0.295	3.11e+06	0.275
batchdes	1.67e+05	0.0487	1.67e+05	0.0148	1.67e+05	0.0278	1.67e+05	0.0676	1.67e+05	0.0273	1.67e+05	0.0322
batchs101006m	7.69e+05	14.5	7.74e+05	0.816	7.69e+05	6.69	7.77e+05	1.29	7.69e+05	6.31	7.69e+05	4.83
batchs121208m	1.24e+06	33.4	1.24e+06	1.7	1.24e+06	13.5	1.24e+06	2.39	1.25e+06	11.8	1.25e+06	5.23
batchs151208m	1.54e+06	170	1.54e+06	1.11	1.54e+06	14.9	1.55e+06	4.47	1.55e+06	13.5	1.56e+06	6.35
batchs201210m	2.3e+06	175	2.3e+06	2.07	2.3e+06	16.4	2.3e+06	6.08	2.3e+06	18.7	2.32e+06	11.9
clay0203hfsg	4.16e+04	9.07	4.16e+04	0.803	4.16e+04	1.23	4.16e+04	0.494	4.17e+04	3.31	4.21e+04	1.61
clay0204hfsg	6.55e+03	49.2	6.54e+03	1.05	6.54e+03	4.31	6.6e+03	1.52	6.54e+03	2.37	6.54e+03	1.68
clay0205hfsg	9.71e+03	300	8.09e+03	14.7	8.09e+03	25.9	8.09e+03	5.18	8.09e+03	7.47	8.09e+03	4.59
clay0303hfsg	2.67e+04	18.6	2.67e+04	0.751	2.67e+04	3	2.67e+04	0.988	2.67e+04	4.21	2.84e+04	2.23
clay0304hfsg	4.03e+04	293	4.03e+04	9.88	4.03e+04	10.4	4.03e+04	4.16	4.03e+04	15.3	4.03e+04	9.81
clay0305hfsg	8.36e+04	300	8.09e+03	14.5	8.09e+03	33.6	8.09e+03	9.91	8.09e+03	13.3	8.09e+03	4.69
cvxnonsep_normcon20	-21.7	0.0234	-21.7	0.0149	-21.7	0.0446	-21.7	0.0625	-21.7	0.0276	-21.7	0.0215
cvxnonsep_normcon30	-34.2	0.0439	-34.2	0.0169	-34.2	0.0203	-34	0.0653	-34.2	0.0317	-34.2	0.0258
cvxnonsep_normcon40	-32.6	0.0612	-32.6	0.0174	-32.6	0.0553	-32.5	0.134	-32.6	0.0377	-32.6	0.0331

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
cvxnonsep_nsig20	80.9	0.0331	81.6	0.0412	81	42.9	81.2	0.0794	80.9	0.0277	80.9	0.0231
cvxnonsep_nsig20r	80.9	0.0268	81.3	0.0463	81.1	0.0705	81	0.267	80.9	0.035	80.9	0.0247
cvxnonsep_nsig30	131	0.0631	131	0.0444	131	300	131	0.0957	131	0.0346	131	0.0303
cvxnonsep_nsig30r	156	0.0638	157	0.0488	157	0.0892	157	0.151	156	0.0389	156	0.0294
cvxnonsep_nsig40	134	0.0908	134	0.0604	138	300	134	0.106	134	0.0418	134	0.0375
cvxnonsep_nsig40r	134	0.0658	134	0.0539	135	0.0721	134	0.148	134	0.214	134	0.173
cvxnonsep_pcon20	-21.5	0.0297	-21.5	124	-21.5	0.0491	-21.5	0.0714	-21.5	0.0305	-21.5	0.0256
cvxnonsep_pcon20r	-21.5	0.0252	-21.4	0.0328	-21.3	0.0183	-21.4	0.107	-21.5	0.0261	-21.5	0.0266
cvxnonsep_pcon30	-36	0.0385	-35.9	300	-36	0.11	-35.9	0.0706	-35.9	0.0378	-35.9	0.0304
cvxnonsep_pcon30r	-36	0.0279	-35.8	0.0475	-36	0.0235	-36	0.136	-35.9	0.0316	-35.9	0.0308
cvxnonsep_pcon40	-46.6	0.0719	-45.3	300	-46.6	0.0671	-46.5	0.0682	-46.6	0.0444	-46.6	0.0397
cvxnonsep_pcon40r	-46.6	0.0426	-46.4	0.0415	-46.2	0.0323	-46.6	0.242	-46.6	0.146	-46.6	0.163
cvxnonsep_psig20	93.8	0.0499	93.9	0.0286	94.1	0.964	93.9	0.112	93.8	0.0254	93.8	0.0319
cvxnonsep_psig20r	95.9	0.0352	95.9	0.28	95.9	0.0981	96.2	0.108	95.9	0.0356	95.9	0.0269
cvxnonsep_psig30	79.1	0.0872	79.6	0.0517	79.1	44.4	79.2	0.101	79	0.0311	79	0.0284
cvxnonsep_psig30r	79	0.0703	79.1	0.0657	79	0.144	79.7	0.208	79	0.0424	79	0.0334
cvxnonsep_psig40	85.7	0.154	85.7	0.0357	86.1	300	85.6	0.123	85.5	0.0371	85.5	0.0351
cvxnonsep_psig40r	NaN	NaN	86.8	0.0557	86.5	0.139	86.5	0.285	86.5	0.14	86.5	0.13
enpro48pb	1.87e+05	6.62	1.87e+05	0.284	1.87e+05	2.32	1.87e+05	0.216	1.87e+05	1.02	1.94e+05	0.749
enpro56pb	2.63e+05	10.3	2.63e+05	0.364	2.63e+05	3.63	2.63e+05	0.359	2.63e+05	0.664	2.63e+05	0.536

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
ex1223	4.58	0.0363	4.58	0.0139	4.58	0.0291	4.58	0.0552	4.58	0.0814	4.58	0.0493
ex1223b	4.58	0.031	4.58	0.0136	4.58	0.0299	4.58	0.0477	4.58	0.0907	4.58	0.0402
fac1	1.61e+08	0.0172	1.61e+08	0.0177	1.61e+08	0.0199	1.61e+08	0.08	1.61e+08	1.91	1.61e+08	2.15
fac2	3.32e+08	0.473	3.32e+08	0.0208	NaN	NaN	3.32e+08	0.107	3.32e+08	0.0567	3.32e+08	0.0497
flay02h	37.9	0.0573	37.9	0.0551	37.9	0.578	37.9	0.0943	37.9	0.0667	37.9	0.0743
flay02m	37.9	0.0251	37.9	0.0356	37.9	0.319	37.9	0.0808	37.9	0.0378	37.9	0.0378
flay03h	49	1.45	49	0.523	49	2.47	49	300	49	1.06	49	0.501
flay03m	49	0.322	49	0.278	49	0.742	49	0.286	49	0.381	49	0.0923
flay04h	54.4	55.3	54.4	2.77	54.4	13.7	54.4	1.25	54.4	8.88	54.4	2.95
flay04m	54.4	8.53	54.4	2.67	54.4	6.21	54.4	0.843	54.4	2.1	54.4	0.3
flay05h	64.5	300	64.5	246	64.5	300	64.5	118	64.5	300	64.5	18.3
flay05m	64.5	300	64.5	140	64.5	262	64.5	62.1	64.5	75.1	64.5	0.944
flay06h	66.9	300	66.9	300	66.9	300	66.9	300	66.9	300	66.9	92.2
flay06m	66.9	300	66.9	300	66.9	300	66.9	300	66.9	300	66.9	4.13
fo7	43.9	300	20.7	88.9	20.7	70.9	20.7	37.9	20.7	149	20.7	11
fo7_2	37.8	300	17.7	30	17.7	54.8	17.7	7.34	17.7	67.6	17.7	3.28
fo7_ar25_1	NaN	NaN	23.1	9.82	23.1	40.7	23.1	34.3	23.1	61.6	25	11.2
fo7_ar2_1	NaN	NaN	24.8	13.3	24.8	24.6	24.8	24.7	24.8	62.6	25	20

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
fo7_ar3_1	NaN	NaN	22.5	28.6	22.5	52.3	22.5	51.3	22.5	57.3	22.5	8.21
fo7_ar4_1	NaN	NaN	20.7	54	20.7	34.1	20.7	61	20.7	76.2	20.7	47
fo7_ar5_1	NaN	NaN	17.7	13.7	17.7	31.4	17.7	28.5	17.7	55.7	17.7	32
fo8	59.7	300	23.9	300	22.4	206	22.4	91.1	22.4	300	24.7	138
fo8_ar25_1	NaN	NaN	28	300	28	156	28	300	28	291	30.8	122
fo8_ar2_1	NaN	NaN	30.3	292	30.3	182	30.3	283	30.3	273	31	60.5
fo8_ar3_1	NaN	NaN	23.9	151	23.9	176	23.9	43.4	23.9	216	23.9	122
fo8_ar4_1	NaN	NaN	22.4	76.2	22.4	300	22.4	44.3	22.4	111	26.4	60.5
fo8_ar5_1	NaN	NaN	22.4	300	22.4	300	22.4	88.2	22.4	144	22.4	60.5
fo9	NaN	NaN	26.8	300	23.5	300	23.5	300	23.5	162	24.4	20.4
fo9_ar25_1	NaN	NaN	32.2	300	32.2	300	32.2	300	32.2	300	32.7	122
fo9_ar2_1	NaN	NaN	32.8	300	32.6	300	32.6	300	32.6	267	32.8	61.1
fo9_ar3_1	NaN	NaN	27.1	300	24.8	122	24.8	59.4	24.8	72.2	24.8	37.5
fo9_ar4_1	NaN	NaN	30.4	300	24.4	300	29.5	300	23.5	74.7	24.8	33.5
fo9_ar5_1	NaN	NaN	27.5	300	28.9	300	24.4	300	24.4	236	24.8	55.5
gams01	2.79e+04	300	2.15e+04	300	2.4e+04	300	2.14e+04	300	2.19e+04	300	2.19e+04	263
ibs2	NaN	NaN	4.45	4.28	4.45	7.08	NaN	NaN	NaN	NaN	NaN	NaN
jit1	1.74e+05	0.051	1.74e+05	0.0275	1.74e+05	0.0894	1.74e+05	0.0764	1.74e+05	0.0274	1.74e+05	0.0214
m3	37.8	0.367	37.8	0.0433	37.8	0.206	37.8	0.0644	37.8	0.112	37.8	0.0756
m6	106	300	82.3	0.226	82.3	0.783	82.3	0.301	82.3	0.898	82.3	0.572
m7	131	300	107	0.383	107	2.65	107	1.02	107	1.65	107	1.12

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
m7_ar25_1	144	300	144	0.376	144	2.4	144	0.594	144	0.788	144	0.633
m7_ar2_1	190	300	190	1.81	190	7.35	190	1.62	190	3.24	190	1.02
m7_ar3_1	NaN	NaN	144	2.36	144	8.13	144	2.17	144	300	156	0.713
m7_ar4_1	NaN	NaN	107	0.972	107	1.92	107	0.985	107	1.66	107	1.19
m7_ar5_1	NaN	NaN	106	2.56	106	6.9	106	3.57	106	6.04	106	1.97
no7_ar25_1	NaN	NaN	108	28.5	108	77.7	108	47.7	108	109	108	60.4
no7_ar2_1	NaN	NaN	108	14.7	108	205	108	21.8	108	67.8	108	60.4
no7_ar3_1	NaN	NaN	108	104	108	300	108	90.4	108	161	108	61
no7_ar4_1	NaN	NaN	98.5	90.2	98.5	167	98.5	86.8	98.5	300	98.9	220
no7_ar5_1	NaN	NaN	90.6	99.1	90.6	118	90.6	120	90.8	300	99.8	61.4
o7	NaN	NaN	135	300	132	300	132	300	NaN	NaN	NaN	NaN
o7_2	161	300	117	300	117	300	117	239	117	300	117	185
o7_ar25_1	NaN	NaN	140	270	140	300	140	185	140	238	140	122
o7_ar2_1	NaN	NaN	140	72.2	140	101	140	63.5	140	159	140	60.5
o7_ar3_1	NaN	NaN	138	300	138	300	138	300	138	300	144	174
o7_ar4_1	NaN	NaN	135	300	135	300	132	300	145	300	137	169
o7_ar5_1	NaN	NaN	119	300	117	300	117	297	124	300	125	181
o8_ar4_1	NaN	NaN	260	300	244	300	255	300	259	300	259	253
o9_ar4_1	NaN	NaN	271	300	327	300	288	300	279	300	NaN	NaN
p_ball_10b_5p_2d_h	18.7	86	18.7	12.3	18.7	15.7	18.7	47.4	18.7	47.5	22	4.63
p_ball_10b_5p_3d_h	NaN	NaN	44	50.4	44	46.9	44	223	44	115	53.9	4.76

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
p_ball_10b_5p_4d_h	71.4	183	85.1	5.04	71.4	104	74.2	300	71.4	300	81.3	5.45
p_ball_10b_7p_3d_h	117	300	110	300	110	300	NaN	NaN	114	300	131	64.4
p_ball_15b_5p_2d_h	6.6	265	6.6	77	6.6	62.1	6.6	270	6.6	121	15.8	4.63
p_ball_20b_5p_2d_h	2.44	300	2.44	100	2.44	140	NaN	NaN	8.08	300	46.3	6.96
p_ball_20b_5p_3d_h	21.8	300	19.7	300	20.5	300	66	300	21.1	300	42.4	38.7
p_ball_30b_10p_2d_h	NaN	NaN	76.8	300	45.4	300	NaN	NaN	NaN	NaN	NaN	NaN
p_ball_30b_5p_2d_h	8.4	300	0.292	248	0.292	300	NaN	NaN	1.54	300	13.3	8.19
p_ball_30b_5p_3d_h	9.14	300	15.7	300	9.14	300	NaN	NaN	16	300	54.2	18.9
p_ball_30b_7p_2d_h	NaN	NaN	13.9	300	14	300	NaN	NaN	22.4	300	30.7	256
p_ball_40b_5p_3d_h	NaN	NaN	108	26.1	12.8	300	NaN	NaN	13.9	300	13.9	18.5
p_ball_40b_5p_4d_h	NaN	NaN	NaN	NaN	54.2	300	NaN	NaN	54	300	81.2	85.5
portfol_buyin	0.0305	0.0437	0.0294	0.0751	0.0294	0.322	0.0294	0.282	0.0305	0.0873	0.0305	0.0787
portfol_card	0.0337	0.0586	0.0322	0.16	0.0322	0.17	0.0322	0.257	0.0322	0.127	0.0322	0.115
portfol_roundlot	NaN	NaN	0.0283	41.3	0.0283	0.0912	0.0283	81.3	0.117	9.6	0.152	24.6
procurement2mot	210	31.6	212	1.04	212	1.36	212	1.98	212	20.1	209	14.3
ravenpb	2.7e+05	1.04	2.7e+05	0.164	2.7e+05	1.91	2.7e+05	300	2.7e+05	1.22	2.7e+05	0.95
risk2bpb	-55.9	5.45	-55.7	0.0353	-55.9	0.0519	-55.7	0.0914	-55.7	4	-55.7	4.39
rsyn0805hfsq	1.3e+03	2.45	1.3e+03	0.0429	1.3e+03	0.191	1.3e+03	0.666	1.3e+03	2.27	1.3e+03	2.1
rsyn0805m	1.29e+03	300	1.3e+03	0.0858	1.3e+03	1.77	1.29e+03	0.154	1.3e+03	1.05	1.28e+03	0.493
rsyn0805m02hfsq	2.24e+03	18.6	2.24e+03	0.312	2.24e+03	0.765	2.24e+03	1.6	2.24e+03	8.25	2.24e+03	9.01

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
rsyn0805m02m	NaN	NaN	2.24e+03	0.303	2.24e+03	10.7	2.24e+03	0.57	2.22e+03	6.65	2.22e+03	3.48
rsyn0805m03hfsq	3.07e+03	27.9	3.07e+03	0.354	3.07e+03	1.19	3.07e+03	2.12	3.07e+03	12.2	3.07e+03	14.2
rsyn0805m03m	NaN	NaN	3.07e+03	0.367	3.07e+03	16	3.06e+03	0.811	3.06e+03	10.1	3.06e+03	6.95
rsyn0805m04hfsq	7.17e+03	0.67	7.14e+03	0.2	7.17e+03	2.3	7.17e+03	2.45	7.17e+03	20.4	7.17e+03	20.6
rsyn0805m04m	NaN	NaN	7.17e+03	0.515	7.17e+03	15.7	7.17e+03	0.637	7.13e+03	10.8	7.13e+03	10.3
rsyn0810hfsq	1.72e+03	2.91	1.72e+03	0.045	1.72e+03	0.106	1.72e+03	0.549	1.72e+03	2.74	1.72e+03	2.81
rsyn0810m	1.64e+03	300	1.72e+03	0.0865	1.72e+03	0.99	1.72e+03	0.141	1.72e+03	1.79	1.7e+03	0.602
rsyn0810m02hfsq	1.74e+03	24	1.74e+03	0.174	1.74e+03	2.33	1.74e+03	2.27	1.73e+03	10.9	1.73e+03	10.5
rsyn0810m02m	NaN	NaN	1.74e+03	0.596	1.74e+03	12.3	1.74e+03	0.605	1.74e+03	9.98	1.7e+03	4.19
rsyn0810m03hfsq	NaN	NaN	2.72e+03	0.713	2.72e+03	4.12	2.72e+03	2.46	2.72e+03	18.4	2.72e+03	14.2
rsyn0810m03m	NaN	NaN	2.72e+03	0.81	2.72e+03	16.6	2.72e+03	1.2	2.72e+03	84	2.65e+03	8.86
rsyn0810m04hfsq	NaN	NaN	6.58e+03	0.294	6.58e+03	3.05	6.58e+03	3.14	6.57e+03	36.4	6.48e+03	22.1
rsyn0810m04m	5.4e+03	300	6.57e+03	0.629	6.58e+03	14.5	6.58e+03	1.13	6.58e+03	21.8	6.49e+03	12.6
rsyn0815hfsq	1.27e+03	4.28	1.27e+03	0.0602	1.27e+03	0.882	1.27e+03	0.692	1.27e+03	2.85	1.27e+03	3.77
rsyn0815m	1.26e+03	300	1.27e+03	0.114	1.27e+03	1.48	1.27e+03	0.157	1.27e+03	2.01	1.23e+03	0.729
rsyn0815m02hfsq	1.77e+03	19.9	1.77e+03	0.305	1.77e+03	3.02	1.77e+03	2.93	1.77e+03	10.6	1.77e+03	12.3
rsyn0815m02m	NaN	NaN	1.77e+03	0.353	1.77e+03	10.9	1.77e+03	0.547	1.76e+03	7.02	1.76e+03	5.34
rsyn0815m03hfsq	2.83e+03	54.3	2.83e+03	0.962	2.83e+03	4.19	2.83e+03	6.39	2.83e+03	17.4	2.83e+03	19.4
rsyn0815m03m	NaN	NaN	2.83e+03	0.688	2.83e+03	22.1	2.83e+03	1.9	2.83e+03	17.5	2.79e+03	8.93
rsyn0815m04hfsq	3.41e+03	82.4	3.41e+03	0.491	3.4e+03	4.47	3.41e+03	10.1	3.41e+03	24.9	3.41e+03	28.3

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
rsyn0815m04m	3.11e+03	300	3.41e+03	0.952	3.41e+03	102	3.41e+03	1.53	3.41e+03	41	3.36e+03	14.7
rsyn0820hfs	1.15e+03	4.61	1.15e+03	0.0843	1.15e+03	0.696	1.15e+03	0.827	1.15e+03	3.14	1.15e+03	3.96
rsyn0820m	1.15e+03	300	1.15e+03	0.165	1.15e+03	1.36	1.15e+03	0.344	1.15e+03	1.67	1.13e+03	0.9
rsyn0820m02hfs	1.09e+03	30.5	1.09e+03	0.3	1.09e+03	2.89	1.09e+03	4.37	1.09e+03	11.8	1.09e+03	11.9
rsyn0820m02m	NaN	NaN	1.09e+03	0.518	1.09e+03	12.7	1.09e+03	0.796	1.09e+03	49.7	1.08e+03	7.24
rsyn0820m03hfs	2.03e+03	95.1	2.03e+03	0.496	2.03e+03	11.1	2.03e+03	9.42	2.01e+03	21.2	2.01e+03	22
rsyn0820m03m	NaN	NaN	2.03e+03	1.43	2.03e+03	23.7	2.03e+03	2.32	2.03e+03	90.5	1.98e+03	12.4
rsyn0820m04hfs	2.45e+03	122	2.45e+03	1.24	2.45e+03	9.31	2.45e+03	19.6	2.44e+03	28.9	2.44e+03	32.6
rsyn0820m04m	947	300	2.45e+03	1.08	2.45e+03	80.8	2.45e+03	3.9	2.45e+03	91.9	2.39e+03	18.4
rsyn0830hfs	510	5.59	509	0.166	510	2.55	510	1.62	509	4.67	509	4.85
rsyn0830m	496	300	510	0.193	510	2.01	510	0.494	510	6.51	496	1.5
rsyn0830m02hfs	731	40.8	728	0.261	729	4.78	731	7.23	728	20.5	721	15.7
rsyn0830m02m	NaN	NaN	731	0.428	731	6.8	731	0.947	729	105	709	8.67
rsyn0830m03hfs	1.54e+03	50.6	1.54e+03	0.431	1.54e+03	9.71	1.54e+03	16.2	1.54e+03	35.5	1.53e+03	27.3
rsyn0830m03m	1.18e+03	300	1.54e+03	0.695	1.54e+03	16.9	1.54e+03	2.49	1.54e+03	45.7	1.5e+03	14.3
rsyn0830m04hfs	2.53e+03	187	2.53e+03	0.499	2.52e+03	15.7	2.53e+03	31	2.52e+03	68.8	2.5e+03	40.1
rsyn0830m04m	706	300	2.53e+03	1.22	2.53e+03	73.2	2.53e+03	5.07	2.52e+03	42.1	2.47e+03	17.6
rsyn0840hfs	326	4.23	326	0.13	326	1.15	326	3.24	326	5.24	326	6.85
rsyn0840m	248	300	326	0.175	326	2.18	326	0.342	326	2.23	326	2.07

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
rsyn0840m02hfsg	735	37.3	735	0.286	735	2.85	735	10.5	735	31.5	720	17.6
rsyn0840m02m	499	300	735	0.551	735	10.3	735	1.75	734	19.3	722	12.7
rsyn0840m03hfsg	2.73e+03	253	2.73e+03	0.464	2.73e+03	4.9	2.74e+03	35.5	2.73e+03	29.8	2.73e+03	34.4
rsyn0840m03m	1.82e+03	300	2.74e+03	0.654	2.74e+03	12.1	2.73e+03	5	2.72e+03	47.9	2.72e+03	47.7
rsyn0840m04hfsg	2.56e+03	189	2.55e+03	0.584	2.56e+03	5.32	2.56e+03	134	2.56e+03	63.4	2.53e+03	44.6
rsyn0840m04m	697	300	2.56e+03	0.668	2.56e+03	173	2.56e+03	10	2.55e+03	114	2.55e+03	113
sssd08-04	1.83e+05	1.47	1.82e+05	0.243	1.82e+05	8.47	1.82e+05	0.186	1.82e+05	0.423	1.82e+05	0.177
sssd12-05	2.82e+05	2.45	2.82e+05	0.409	2.81e+05	300	2.82e+05	0.221	2.82e+05	0.85	2.82e+05	0.443
sssd15-04	2.05e+05	1.52	2.05e+05	0.321	2.05e+05	300	2.05e+05	0.234	2.06e+05	0.787	2.06e+05	0.404
sssd15-06	5.44e+05	18.7	5.44e+05	0.563	5.42e+05	300	5.42e+05	0.626	5.42e+05	7.27	5.44e+05	0.558
sssd15-08	5.67e+05	123	5.67e+05	0.587	5.63e+05	300	5.65e+05	0.859	5.63e+05	4.75	5.63e+05	1.74
sssd16-07	4.2e+05	19.4	4.2e+05	0.547	4.19e+05	300	4.2e+05	0.692	4.2e+05	4.66	4.2e+05	1.3
sssd18-06	4.01e+05	4.52	4e+05	0.326	3.98e+05	300	3.99e+05	0.545	3.99e+05	2.18	3.99e+05	0.82
sssd18-08	8.4e+05	88.4	8.36e+05	0.565	8.38e+05	300	8.37e+05	1.32	8.35e+05	5	8.35e+05	1.26
sssd20-04	3.5e+05	2.09	3.48e+05	0.168	3.48e+05	300	3.49e+05	0.236	3.48e+05	0.822	3.48e+05	0.485
sssd20-08	4.73e+05	41.5	4.72e+05	0.438	4.72e+05	300	4.95e+05	0.863	4.7e+05	2.86	4.7e+05	2.03
sssd22-08	5.12e+05	36.1	5.11e+05	0.407	5.09e+05	300	5.12e+05	0.988	5.09e+05	7.76	5.09e+05	6.66
sssd25-04	3.01e+05	3.05	3.01e+05	0.38	3e+05	300	3.02e+05	0.214	3e+05	0.575	3e+05	0.591
sssd25-08	4.76e+05	24.7	4.76e+05	0.7	4.74e+05	300	4.74e+05	1.12	4.73e+05	5.13	4.73e+05	2.92
st_el4	4.58	0.0405	4.58	0.0275	4.58	0.0279	4.58	0.0635	4.58	0.0985	4.58	0.0529
stockcycle	1.2e+05	19.9	1.2e+05	0.19	1.2e+05	2.22	1.21e+05	0.273	1.3e+05	300	1.31e+05	300

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
syn05hfsq	837	0.0138	836	0.0339	838	0.036	838	0.121	837	0.0615	837	0.0623
syn05m	838	0.0472	838	0.0279	837	0.0165	837	0.0669	838	0.0555	838	0.0599
syn05m02hfsq	3.03e+03	0.0229	3.03e+03	0.0578	3.03e+03	0.0371	3.03e+03	0.317	3.03e+03	0.149	3.03e+03	0.166
syn05m02m	3.03e+03	0.218	3.02e+03	0.0332	3.03e+03	0.0313	3.03e+03	0.0751	3.03e+03	0.123	3.03e+03	0.098
syn05m03hfsq	4.03e+03	0.0313	4.03e+03	0.0613	4.03e+03	0.0449	4.02e+03	0.276	4.03e+03	0.276	4.03e+03	0.298
syn05m03m	4.03e+03	0.634	3.99e+03	0.0363	3.99e+03	0.0407	4.03e+03	0.0811	4.03e+03	0.217	4.03e+03	0.316
syn05m04hfsq	5.51e+03	0.0389	5.51e+03	0.0773	5.51e+03	0.07	5.51e+03	0.703	5.51e+03	0.401	5.51e+03	0.399
syn05m04m	5.51e+03	1.28	5.49e+03	0.0424	5.51e+03	0.0837	5.51e+03	0.0866	5.51e+03	0.387	5.51e+03	0.331
syn10hfsq	1.27e+03	0.0168	1.27e+03	0.0348	1.27e+03	0.0374	1.27e+03	0.211	1.27e+03	0.191	1.27e+03	0.216
syn10m	1.27e+03	0.233	1.27e+03	0.0317	1.27e+03	0.0324	1.27e+03	0.0671	1.27e+03	0.187	1.25e+03	0.0953
syn10m02hfsq	2.31e+03	0.0462	2.31e+03	0.0523	2.31e+03	0.0726	2.31e+03	0.384	2.31e+03	0.381	2.31e+03	0.312
syn10m02m	2.31e+03	3.42	2.31e+03	0.0446	2.31e+03	0.24	2.31e+03	0.112	2.31e+03	0.243	2.31e+03	0.211
syn10m03hfsq	3.35e+03	0.0716	3.34e+03	0.104	3.35e+03	0.0736	3.35e+03	0.481	3.35e+03	1.28	3.35e+03	1.37
syn10m03m	NaN	NaN	3.34e+03	0.0519	3.35e+03	0.454	3.35e+03	0.125	3.35e+03	0.352	3.35e+03	0.293
syn10m04hfsq	4.56e+03	0.111	4.54e+03	0.0791	4.55e+03	0.108	4.56e+03	0.792	4.56e+03	2.57	4.56e+03	2.51
syn10m04m	NaN	NaN	4.53e+03	0.0509	4.56e+03	0.53	4.56e+03	0.134	4.56e+03	0.564	4.56e+03	0.538
syn15hfsq	853	0.0228	853	0.0429	845	0.0993	853	0.248	853	0.228	853	0.234
syn15m	853	0.842	853	0.0303	845	0.0603	853	0.0806	853	0.154	853	0.138
syn15m02hfsq	2.83e+03	0.061	2.82e+03	0.069	2.82e+03	0.224	2.83e+03	0.646	2.82e+03	1.9	2.82e+03	1.95
syn15m02m	2.83e+03	9.48	2.83e+03	0.038	2.83e+03	0.205	2.83e+03	0.101	2.83e+03	0.343	2.83e+03	0.276

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
syn15m03hfsq	3.85e+03	0.0732	3.83e+03	0.101	3.85e+03	0.744	3.85e+03	1.22	3.85e+03	3.7	3.85e+03	3.41
syn15m03m	3.85e+03	54.2	3.83e+03	0.0572	3.82e+03	0.196	3.84e+03	0.1119	3.85e+03	1.52	3.85e+03	1.46
syn15m04hfsq	4.94e+03	0.113	4.94e+03	0.134	4.89e+03	0.188	4.94e+03	3.16	4.94e+03	5.1	4.94e+03	5.31
syn15m04m	NaN	NaN	4.92e+03	0.0305	4.94e+03	0.954	4.94e+03	0.133	4.92e+03	3.36	4.92e+03	3.13
syn20hfsq	924	0.0255	918	0.0271	922	0.422	920	1.66	924	0.256	924	0.255
syn20m	924	5.72	923	0.0198	924	0.456	924	0.0843	924	0.214	909	0.0917
syn20m02hfsq	1.75e+03	0.111	1.75e+03	0.0483	1.75e+03	0.687	1.75e+03	0.875	1.75e+03	2.64	1.75e+03	2.34
syn20m02m	NaN	NaN	1.75e+03	0.0253	1.75e+03	0.347	1.75e+03	0.1	1.75e+03	0.852	1.75e+03	0.452
syn20m03hfsq	2.65e+03	0.188	2.64e+03	0.0718	2.65e+03	0.528	2.65e+03	2.62	2.63e+03	5.01	2.63e+03	4.72
syn20m03m	NaN	NaN	2.63e+03	0.0358	2.65e+03	0.859	2.65e+03	0.217	2.65e+03	3.18	2.6e+03	1.84
syn20m04hfsq	3.53e+03	0.316	3.52e+03	0.0925	3.53e+03	2.11	3.53e+03	5.48	3.52e+03	4.15	3.52e+03	8.25
syn20m04m	3.51e+03	300	3.52e+03	0.0455	3.53e+03	1.17	3.53e+03	0.304	3.53e+03	5.16	3.49e+03	4.39
syn30hfsq	138	0.623	138	0.0944	138	1.33	131	2.05	138	0.79	138	0.722
syn30m	NaN	NaN	138	0.0733	138	0.269	138	0.23	138	0.701	125	0.172
syn30m02hfsq	400	2.18	400	0.215	400	2.95	400	3.5	400	9.92	387	4.61
syn30m02m	NaN	NaN	400	0.0839	400	1.19	400	0.269	400	3.03	391	1.88
syn30m03hfsq	654	11.4	654	0.4	654	4.05	654	8.31	654	11.3	654	8.74
syn30m03m	NaN	NaN	654	0.0815	654	3.55	654	0.491	654	5.78	633	3.53

Table 7 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
syn30m04hfg	866	22.7	864	0.531	866	4.12	866	18.6	866	15.9	866	13.2
syn30m04m	787	300	865	0.198	866	3.7	864	1.02	866	14.7	822	6.58
syn40hfg	67.7	1.1	67.7	0.206	67.7	2.23	64.7	2.75	67.7	1.88	67.7	1.89
syn40m	66.8	300	67.7	0.172	67.7	1.98	67.7	0.358	67.7	0.748	66.8	0.358
syn40m02hfg	389	3.76	389	0.258	387	2.81	389	3.26	388	11.4	387	8.89
syn40m02m	NaN	NaN	387	0.169	389	1.9	388	0.417	387	6.4	376	4.48
syn40m03hfg	395	21.3	395	1.32	395	4.3	395	17.8	390	19.9	382	16.3
syn40m03m	287	300	395	0.383	395	3.16	393	1.35	393	15.5	377	11.4
syn40m04hfg	902	111	901	0.444	902	2.54	902	49	902	19.3	902	20.3
syn40m04m	769	300	899	0.319	902	7.04	902	3.58	902	19	862	12
synthes1	6.01	0.0354	6.01	0.0136	6.01	0.0898	6.01	0.0754	6.01	0.0365	6.01	0.042
synthes2	73	0.058	73	0.0143	73	0.0216	73	0.0723	73	0.0871	73	0.0416
synthes3	68	0.103	68.1	0.0139	68.1	0.0528	68	0.0819	68	0.103	68	0.102
tls12	NaN	NaN	NaN	NaN	NaN	NaN	106	300	NaN	NaN	NaN	NaN
tls2	5.3	7.25	5.3	0.0532	5.3	0.369	5.3	0.104	5.3	0.446	5.3	0.428
tls4	NaN	NaN	8.3	6.86	8.3	25.3	8.3	20.4	8.3	300	22	141
tls5	NaN	NaN	10.3	300	10.7	300	10.3	300	NaN	NaN	NaN	NaN
tls6	NaN	NaN	15.3	300	17.1	300	15.6	300	NaN	NaN	NaN	NaN
tls7	NaN	NaN	16.3	300	21.9	300	15.7	300	NaN	NaN	NaN	NaN

Table 8 Benchmark results for nonconvex MINLPs

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
4stufen	1.17e+05	1.7	1.16e+05	300	8.32e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
autocorr_bem20-05	-408	0.0147	-368	0.0509	-337	0.0509	-416	4.6	-384	0.04	-384	0.0351
autocorr_bem20-10	-2.89e+03	0.186	-2.69e+03	0.223	-2.82e+03	0.225	-2.94e+03	66	-2.93e+03	0.207	-2.93e+03	0.193
autocorr_bem20-15	-5.82e+03	0.248	-5.05e+03	0.457	-364	0.487	-5.96e+03	183	-5.82e+03	0.438	-5.82e+03	0.43
autocorr_bem25-06	-928	0.039	-592	0.09	-824	0.0902	-960	97.8	-944	0.0779	-944	0.0656
autocorr_bem25-13	-8.06e+03	0.614	NaN	NaN	-264	0.391	-8.14e+03	300	-7.99e+03	0.365	-7.99e+03	0.348
autocorr_bem25-19	-1.45e+04	1.54	-1.1e+04	0.919	-612	0.938	-1.46e+04	300	-1.45e+04	0.897	-1.45e+04	0.904
autocorr_bem25-25	-1.04e+04	1.8	-9.78e+03	1.01	-1.1e+03	1.03	-1.05e+04	300	-1.05e+04	0.982	-1.05e+04	0.943
autocorr_bem30-04	-320	0.0196	-280	0.0645	-252	0.063	-324	5.1	-320	0.0516	-320	0.0427
autocorr_bem30-08	-2.9e+03	0.26	-2.7e+03	0.217	-2.27e+03	0.224	-2.95e+03	300	-2.9e+03	0.204	-2.9e+03	0.233
autocorr_bem30-15	-1.54e+04	1.02	-1.36e+04	0.845	-772	0.862	-1.57e+04	300	-1.54e+04	0.821	-1.54e+04	0.808
autocorr_bem30-23	-3.02e+04	2.41	-2.87e+04	1.8	-924	1.81	-3.03e+04	300	-3.03e+04	1.73	-3.03e+04	1.76
autocorr_bem30-30	-2.26e+04	2.74	NaN	NaN	-2.88e+03	1.88	-2.25e+04	300	-2.25e+04	1.8	-2.25e+04	1.94
autocorr_bem35-04	-384	0.0153	-340	0.0592	-264	0.0616	-384	9.57	-368	0.0482	-368	0.0426
autocorr_bem35-09	-5.05e+03	0.455	-4.64e+03	0.452	-3.52e+03	0.458	-5.1e+03	300	-5.08e+03	0.436	-5.08e+03	0.432
autocorr_bem35-18	-3.11e+04	1.78	-2.99e+04	2.45	-544	2.48	-3.11e+04	300	-3.09e+04	2.42	-3.09e+04	2.42
autocorr_bem35-26	-5.49e+04	3.6	NaN	NaN	-1.2e+03	2.78	-5.5e+04	300	-5.51e+04	2.66	-5.51e+04	2.59
autocorr_bem35-35fix	-4.02e+04	5.91	-1.65e+04	3.77	-4e+03	3.85	-4.02e+04	300	-4.05e+04	3.71	-4.05e+04	3.8
autocorr_bem40-05	-912	0.0335	-600	0.0755	-744	0.0767	-936	300	-896	0.0644	-896	0.0629
autocorr_bem40-10	-8.13e+03	0.701	NaN	NaN	-144	0.319	-8.19e+03	300	-8.14e+03	0.29	-8.14e+03	0.334
autocorr_bem40-20	-5.01e+04	1.89	NaN	NaN	-684	2.29	-5.02e+04	300	-5.02e+04	2.2	-5.02e+04	2.37

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
autocorr_bem40-30	-9.43e+04	6.61	-3.02e+04	4.18	NaN	NaN	-9.46e+04	300	-9.45e+04	4.12	-9.45e+04	3.92
autocorr_bem40-40	-6.67e+04	5.13	-5.13e+04	4.75	NaN	NaN	-6.59e+04	300	-6.74e+04	4.66	-6.74e+04	4.77
autocorr_bem45-05	-1.05e+03	0.151	-764	0.0911	-848	0.0915	-1.06e+03	300	-1.02e+03	0.0784	-1.02e+03	0.0645
autocorr_bem45-11	-1.26e+04	0.918	-1.1e+04	0.677	-180	0.691	-1.26e+04	300	-1.25e+04	0.653	-1.25e+04	0.677
autocorr_bem45-23	-8.48e+04	4.62	-7.88e+04	3.85	-924	3.94	-8.49e+04	300	-8.47e+04	3.79	-8.47e+04	3.73
autocorr_bem45-34	-1.51e+05	5.42	-1.37e+05	6.78	NaN	NaN	-1.51e+05	300	-1.52e+05	6.69	-1.52e+05	6.63
autocorr_bem45-45	-1.11e+05	19.3	NaN	NaN	NaN	NaN	-1.08e+05	300	-1.12e+05	7.02	-1.12e+05	9.89
autocorr_bem50-06	-2.11e+03	0.495	-464	0.158	-40	0.161	-2.16e+03	300	-2.06e+03	0.145	-2.06e+03	0.125
autocorr_bem50-13	-2.34e+04	2.08	-2.19e+04	1.48	-564	1.49	-2.35e+04	300	-2.36e+04	1.43	-2.36e+04	1.48
autocorr_bem50-25	-1.24e+05	4.12	-1.01e+05	4.37	NaN	NaN	-1.24e+05	300	-1.24e+05	4.28	-1.24e+05	4.16
autocorr_bem50-38	-2.31e+05	11.9	-1.78e+05	10.8	-2.66e+03	11	-2.25e+05	300	-2.32e+05	10.6	-2.32e+05	10.7
autocorr_bem50-50	-1.66e+05	24.8	NaN	NaN	NaN	NaN	-1.63e+05	300	-1.67e+05	10.4	-1.67e+05	9.92
autocorr_bem55-06	-2.3e+03	0.455	-2.04e+03	0.275	-2.1e+03	0.279	-2.4e+03	300	-2.3e+03	0.261	-2.3e+03	0.175
autocorr_bem55-14	-3.29e+04	2.89	-2.92e+04	1.8	-312	1.82	-3.29e+04	300	-3.28e+04	1.75	-3.28e+04	1.6
autocorr_bem55-28	-1.9e+05	5.73	-1.55e+05	8.33	NaN	NaN	-1.88e+05	300	-1.89e+05	8.22	-1.89e+05	7.42
autocorr_bem55-41	-3.36e+05	20.9	-8.9e+03	13.3	NaN	NaN	-3.3e+05	300	-3.37e+05	13.1	-3.37e+05	12.9
autocorr_bem55-55	-2.4e+05	31.5	NaN	NaN	NaN	NaN	-2.28e+05	300	-2.38e+05	16.3	-2.38e+05	15.1
autocorr_bem60-08	-6.65e+03	0.936	-1.45e+03	0.475	-84	0.483	-6.74e+03	300	-6.73e+03	0.455	-6.73e+03	0.457
autocorr_bem60-15	-4.45e+04	2.23	-3.76e+04	2.4	-364	2.45	-4.46e+04	300	-4.44e+04	2.36	-4.44e+04	2.42
autocorr_bem60-30	-2.59e+05	11.1	NaN	NaN	NaN	NaN	-2.58e+05	300	-2.59e+05	7.56	-2.59e+05	6.76

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
autocorr_bem60-45	-4.78e+05	32	NaN	NaN	NaN	NaN	-4.67e+05	300	-4.79e+05	17.7	-4.79e+05	22
autocorr_bem60-60	-3.43e+05	30.5	NaN	NaN	-6.84e+03	24.5	-3.32e+05	300	-3.51e+05	23.6	-3.51e+05	23.3
batch0812_nc	2.69e+06	4.58	2.69e+06	0.0565	2.69e+06	0.586	2.69e+06	0.281	2.71e+06	0.769	2.71e+06	0.535
batch_nc	2.86e+05	0.473	2.86e+05	0.0745	4.6e+04	0.0745	2.86e+05	0.326	2.86e+05	0.0644	2.86e+05	0.118
beuster	NaN	NaN	1.17e+05	300	1.21e+05	300	NaN	NaN	NaN	NaN	NaN	NaN
cascetanks	9.16	5.43	9.18	8.22	251	8.22	9.16	11.6	9.16	8.2	9.16	10.6
case_1scv2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cecil_13	-1.16e+05	100	-1.16e+05	0.234	-1.16e+05	11.9	-1.1e+05	12.2	-9.48e+04	18	-9.34e+04	14.6
chp_partload	NaN	NaN	24.7	211	20	300	NaN	NaN	NaN	NaN	NaN	NaN
chp_shorttermplan1a	NaN	NaN	215	0.483	215	4.01	341	300	217	259	NaN	NaN
chp_shorttermplan1b	273	189	254	72.8	255	300	NaN	NaN	NaN	NaN	NaN	NaN
chp_shorttermplan2a	NaN	NaN	2.46e+05	8.35	2.47e+05	20.7	NaN	NaN	NaN	NaN	NaN	NaN
chp_shorttermplan2b	NaN	NaN	-1.63e+05	5.53	-1.63e+05	245	NaN	NaN	NaN	NaN	NaN	NaN
chp_shorttermplan2c	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
chp_shorttermplan2d	NaN	NaN	4.87e+05	300	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
contrvar	8.09e+05	7.98	8.28e+05	300	8.16e+05	300	8.09e+05	300	8.1e+05	300	8.09e+05	300
csched1	-3.06e+04	0.763	-3.06e+04	0.208	-2.97e+04	0.24	NaN	NaN	-2.93e+04	0.229	-2.93e+04	0.217
csched1a	-3.04e+04	0.103	-2.99e+04	0.04	-2.93e+04	0.0405	-3.03e+04	0.1	-3.03e+04	0.0305	-3.03e+04	0.0436
csched2	-1.66e+05	7.54	-1.66e+05	65.8	-1.42e+05	300	NaN	NaN	-1.58e+05	300	-1.63e+05	300

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
csched2a	-1.65e+05	3.2	-1.65e+05	3.84	-1.61e+05	93.1	-1.43e+05	0.729	-1.59e+05	92.9	-1.58e+05	3.04
deb10	209	0.723	NaN	NaN	NaN	NaN	NaN	NaN	209	0.51	209	0.506
deb6	202	180	NaN	NaN	242	16.9	NaN	NaN	202	16.9	205	5.85
deb7	117	113	NaN	NaN	NaN	NaN	NaN	NaN	119	300	186	133
deb8	NaN	NaN	NaN	NaN	4.37e+03	289	NaN	NaN	117	289	8.69e+03	132
deb9	117	121	NaN	NaN	NaN	NaN	NaN	NaN	120	300	176	133
eg_all_s	7.92	144	NaN	NaN	9.41	40.9	9.77	59.3	9.15	40.6	10	26
eg_disc2_s	5.64	56	5.89	2.15	6.71	33	5.67	300	5.64	32.6	5.64	22.2
eg_disc_s	5.76	33.6	6.19	0.991	7.96	19.6	6.41	75.8	5.76	19.2	5.76	8.97
eg_int_s	7.89	32.7	11.7	70.8	7.46	213	8.39	300	7.89	213	8.03	300
emiplac	NaN	NaN	-1.31e+05	0.029	-1.32e+05	0.333	-1.28e+05	2.18	NaN	NaN	NaN	NaN
ex1221	7.67	0.00655	7.67	0.00992	7.67	0.00995	7.67	0.0795	7.67	0.0219	7.67	0.0249
ex1222	1.08	0.0176	1.08	0.0107	1.08	0.00932	1.08	0.0533	1.08	0.0229	1.08	0.015
ex1224	-0.943	0.0567	-0.943	0.0184	-0.943	0.0273	0	0.109	-0.905	0.0407	-0.905	0.0273
ex1225	31	0.127	31	0.0112	31	0.0106	34	0.087	31	0.025	31	0.0163
ex1226	-17	0.00827	-17	0.0113	-17	0.0146	-17	0.0724	-17	0.0178	-17	0.0148
ex1233	1.55e+05	1.04	1.56e+05	0.458	1.98e+05	0.458	1.92e+05	0.887	1.56e+05	0.447	1.94e+05	0.214
ex1243	8.34e+04	2.07	8.34e+04	0.0843	1.91e+05	0.464	1.29e+05	0.266	9.85e+04	0.453	9.85e+04	0.925
ex1244	NaN	NaN	8.2e+04	0.127	8.2e+04	1.43	8.4e+04	0.496	8.2e+04	1.42	8.36e+04	0.568
ex1252	1.29e+05	0.837	1.32e+05	0.515	1.64e+05	0.517	1.34e+05	0.838	1.34e+05	0.505	1.53e+05	0.137
ex1252a	NaN	NaN	1.34e+05	0.251	1.34e+05	0.252	1.34e+05	0.807	1.34e+05	0.242	1.51e+05	0.121

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
ex3pb	68	0.0866	68.7	0.0136	68	0.0455	68	66.6	68	0.38	76.4	0.0398
feedtray	-13.4	0.0435	-37.5	0.279	-37.5	0.256	NaN	NaN	-13.4	0.241	-13.4	0.204
fin2bb	1.82e-08	236	730	23.4	NaN	NaN	0	27.2	9.74e-17	23.4	9.74e-17	17.7
fuzzy	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gams02	NaN	NaN	9.93e+07	300	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gams04	NaN	NaN	4.18e+05	1.11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gasnet	NaN	NaN	7.01e+06	300	7.13e+06	300	NaN	NaN	NaN	NaN	NaN	NaN
gastrans	89.1	0.0465	89.1	0.0213	89.1	0.0422	NaN	NaN	89.1	0.305	89.1	0.348
gastrans040	NaN	NaN	NaN	NaN	NaN	NaN	0	1.22	0	1.39	0	1.98
gastrans135	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	75.9	0	80
gastrans582_cold13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_cold13_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_cold17	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_cold17_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_cool12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_cool12_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_cool14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_cool14_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_freezing27	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_freezing27_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
gastrans582_freezing30	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_freezing30_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_mild10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_mild10_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_mild11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_mild11_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_warm15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_warm15_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_warm31	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gastrans582_warm31_95	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gear4	1.64	59.8	1.64	0.0915	31.5	0.169	946	0.135	165	0.159	235	0.0808
ghg_1veh	7.78	0.0339	7.87	0.0585	7.8	0.06	NaN	NaN	7.78	0.0487	7.78	0.0499
ghg_2veh	7.78	0.232	7.84	0.214	8.06	0.216	NaN	NaN	7.77	0.204	7.77	0.2
ghg_3veh	NaN	NaN	7.8	1.33	7.78	1.34	NaN	NaN	7.78	1.32	7.78	1.67
gkocis	-1.92	0.0329	-1.92	0.0122	-1.92	0.0184	-1.92	0.098	-1.92	0.0437	-1.72	0.0211
hadamard_4	-9.99e-12	0.0166	3	0.0212	3	0.0209	3	0.0951	7.49e-09	0.012	7.49e-09	0.0141
hadamard_5	-9.99e-12	0.0489	5	0.0276	-8.88e-16	0.0279	5	13.2	7.49e-09	0.0178	7.49e-09	0.0206
hadamard_6	-9.99e-12	0.834	8	0.0996	3	0.102	9	300	7.49e-09	0.0867	7.49e-09	0.1
hadamard_7	-9.99e-12	14.9	NaN	NaN	NaN	NaN	24	300	7.49e-09	0.822	7.49e-09	0.903
hadamard_8	-9.99e-12	191	NaN	NaN	3	11.9	24	299	7.49e-09	10.3	7.49e-09	10.7

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
hadamard_9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
hda	-5.96e+03	276	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
heatexch_gen1	NaN	NaN	1.93e+05	5	NaN	NaN	1.89e+05	79.1	5.95e+05	4.99	5.95e+05	5.4
heatexch_gen2	NaN	NaN	1.04e+06	300	1.81e+04	300	1.36e+06	64.8	NaN	NaN	NaN	NaN
heatexch_gen3	NaN	NaN	783	300	783	300	NaN	NaN	NaN	NaN	NaN	NaN
heatexch_spec1	1.55e+05	0.6	5.95e+05	0.32	2.78e+04	0.321	1.68e+05	0.614	1.55e+05	0.31	1.67e+05	0.143
heatexch_spec2	6.35e+05	3.91	6.85e+05	0.68	4.6e+04	0.681	6.62e+05	1.19	6.35e+05	0.67	6.35e+05	0.901
heatexch_spec3	NaN	NaN	6.41e+04	300	6.44e+04	300	4.75e+05	6.47	NaN	NaN	NaN	NaN
heatexch_trigen	NaN	NaN	1e+06	0.284	9.98e+05	1.38	NaN	NaN	NaN	NaN	NaN	NaN
hybridynamic_var	1.54	0.124	1.54	0.0324	1.54	0.133	NaN	NaN	1.54	0.221	1.54	0.131
johmall	-225	0.764	-225	1.59	NaN	NaN	NaN	NaN	-225	3.06	-225	2.37
kan_peaks_h1_n2_g24	2.47	5.03	-5.2	1.16	NaN	NaN	NaN	NaN	0.715	15.1	0.715	15.3
kan_peaks_h1_n2_g3	-2.31	0.3	-3.72	0.197	NaN	NaN	NaN	NaN	-3.19	1.15	-3.19	1.1
kan_peaks_h1_n5	NaN	NaN	-4.96	49.6	NaN	NaN	-3.02	30.4	0.358	49.6	0.358	43.4
kan_r3_h1_n3	NaN	NaN	1.11	19.7	NaN	NaN	9.55	13.8	1.3	19.7	1.3	20.5
kan_r3_h1_n4	202	7.64	0.0766	300	89.2	300	NaN	NaN	NaN	NaN	NaN	NaN
kan_r3_h1_n5	36.5	5.99	0.00143	300	2.53	300	NaN	NaN	NaN	NaN	NaN	NaN
kan_r3_h1_n9	237	16.6	1.21e+03	105	NaN	NaN	NaN	NaN	0.628	105	0.628	103
kan_r5_h1_n3	NaN	NaN	232	17.8	NaN	NaN	765	93.7	65.1	17.8	65.1	17.2
kan_r5_h1_n5	NaN	NaN	50.7	30.4	NaN	NaN	NaN	NaN	1.92	30.4	1.92	32.6
kan_r5_h1_n8	8.18	19.3	1.08e+03	66.7	NaN	NaN	NaN	NaN	49.5	66.7	49.5	57.3

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
kport20	NaN	NaN	31.9	37	31.8	138	NaN	NaN	34.8	300	32.9	84.7
kport40	NaN	NaN	39.6	4.42	40.7	4.43	NaN	NaN	37.6	4.41	37.6	2.33
lip	NaN	NaN	5.68e+06	0.0389	5.68e+06	0.291	5.64e+06	0.97	5.51e+06	0.28	5.51e+06	0.264
mbid	NaN	NaN	4.67	139	9.42	236	4.67	299	9.67	235	10.3	137
multinfract	NaN	NaN	2.63	3.74	2.63	237	NaN	NaN	NaN	NaN	NaN	NaN
multiplants_mtg1a	392	300	386	4.4	227	4.4	NaN	NaN	392	4.39	386	2.71
multiplants_mtg1b	451	300	428	11.7	42.8	11.7	NaN	NaN	450	11.7	391	6.11
multiplants_mtg1c	NaN	NaN	466	5.06	57.4	5.07	NaN	NaN	684	5.05	684	4.61
multiplants_mtg2	7.09e+03	174	6.95e+03	1.03	NaN	NaN	NaN	NaN	7.09e+03	3.95	7.09e+03	3.98
multiplants_mtg5	5.9e+03	89.4	5.92e+03	7.07	5.65e+03	18.9	NaN	NaN	5.92e+03	18.9	5.91e+03	28
multiplants_mtg6	5.05e+03	300	5.05e+03	1.61	5.2e+03	88.7	NaN	NaN	5.27e+03	88.6	5.27e+03	84.6
multiplants_stg1	323	300	338	115	298	116	NaN	NaN	355	115	326	35.7
multiplants_stg1a	173	300	369	214	356	214	NaN	NaN	391	214	342	22.9
multiplants_stg1b	437	300	306	59.1	50.7	59.1	NaN	NaN	468	59.1	468	34.2
multiplants_stg1c	687	300	676	50.7	NaN	NaN	109	5.67	698	50.7	698	12.7
multiplants_stg5	NaN	NaN	5.46e+03	97.1	4.75e+03	97.1	3.9e+03	8.47	5.83e+03	97.1	5.79e+03	21.9
multiplants_stg6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.17e+03	115	5.17e+03	113
nvs01	12.5	0.0237	12.5	0.03	NaN	NaN	NaN	NaN	12.5	0.0437	12.5	0.0523
nvs05	NaN	NaN	NaN	NaN	19.8	0.38	NaN	NaN	16.8	0.37	16.8	0.536
nvs08	23.4	0.0248	23.4	0.0245	23.4	0.0394	23.4	0.0863	23.4	0.03	23.4	0.0354
nvs21	-5.68	0.101	-5.68	0.0136	-5.68	0.0247	-0.00651	0.245	-4.95	0.0588	-4.95	0.0701

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
nvs22	6.06	0.104	24.9	0.0946	62.6	0.0527	NaN	NaN	6.06	0.0857	6.06	0.0965
oae	-1.92	0.0214	-1.92	0.0117	-1.92	0.0187	-1.92	0.104	NaN	NaN	NaN	NaN
oil	-0.851	300	-0.927	300	-0.759	300	NaN	NaN	NaN	NaN	NaN	NaN
oil2	-0.733	1.07	-0.411	3.3	-0.733	300	NaN	NaN	NaN	NaN	NaN	NaN
ortez	-9.53e+03	5.12	-9.53e+03	0.0166	-9.53e+03	0.361	-9.53e+03	3.2	NaN	NaN	NaN	NaN
parallel	924	0.823	924	0.404	925	1.9	2.64e+03	0.51	924	1.88	924	1.84
pooling_epa1	NaN	NaN	NaN	NaN	-281	2	NaN	NaN	-281	1.98	-281	1.97
pooling_epa2	NaN	NaN	-4.35e+03	20.1	-3.52e+03	20.1	NaN	NaN	-3.43e+03	20.1	-3.43e+03	19.3
pooling_epa3	-1.49e+04	300	-1.49e+04	35.7	NaN	NaN	NaN	NaN	-1.4e+04	55.8	-1.14e+04	92.6
primary	NaN	NaN	-18.4	300	-0.863	300	NaN	NaN	NaN	NaN	NaN	NaN
procsel	-1.92	0.0337	-1.92	0.0136	-1.92	0.0181	-1.92	0.088	-1.41	0.0199	-1.41	0.0154
saa_2	12.7	300	12.9	300	NaN	NaN	NaN	NaN	12.2	300	12.7	286
sepasequ_complex	457	300	430	4.9	403	59.1	NaN	NaN	369	59	457	36.3
sepasequ_conv	491	300	482	1.29	491	19.7	NaN	NaN	430	19.6	430	21.1
sfacloc1_2_80	12.8	38.5	12.8	5.04	15.1	6.98	15.8	11.3	12.9	5.02	13.1	4.17
sfacloc1_2_90	18	0.095	18	0.978	18.3	0.982	18.9	4.5	18	0.968	18	0.761
sfacloc1_2_95	18.9	0.0732	18.9	0.792	18.9	0.795	NaN	NaN	18.9	0.781	18.9	0.787
sfacloc1_3_80	8.52	44.1	8.74	8.64	8.94	8.66	NaN	NaN	8.61	8.62	8.61	11
sfacloc1_3_90	11.9	0.104	12	1.63	13.8	1.65	NaN	NaN	11.9	1.62	11.9	5.89
sfacloc1_3_95	21.5	0.301	12.7	1.39	12.5	1.4	NaN	NaN	12.5	1.38	12.5	1.48
sfacloc1_4_80	NaN	NaN	8.26	23.9	8.15	23.9	NaN	NaN	8.08	23.9	8.09	17.7
sfacloc1_4_90	10.6	0.538	12	2.65	14.6	2.66	29.8	19.4	11.1	2.64	11.1	4.06

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
sfacloc1_4_95	11.3	0.49	12.6	3.13	11.3	3.13	NaN	NaN	11.3	3.11	11.3	3.08
sfacloc2_2_80	13.8	300	13.3	2.5	20.3	4.31	23.3	15	13.3	4.29	13.3	4.06
sfacloc2_2_90	18.7	21.5	18.6	0.541	26.8	1.15	18.8	2.37	18.8	1.13	18.8	1.42
sfacloc2_2_95	19.6	12.4	19.6	0.493	29.7	0.84	23.7	2.82	19.6	0.825	19.6	0.983
sfacloc2_3_80	23.7	300	11.3	6.75	19.3	6.97	15.5	19	11.3	6.72	11.5	4.11
sfacloc2_3_90	15.1	69.6	15.1	2.2	20.3	2.2	23.9	1.48	15.8	2.18	15.8	1.95
sfacloc2_3_95	16.2	34.6	16.2	3.35	16.2	3.97	21	5.07	16.7	4.36	16.7	5.22
sfacloc2_4_80	NaN	NaN	10.2	11.2	11.6	11.2	19.7	5.28	13.3	11.1	12.6	7.92
sfacloc2_4_90	NaN	NaN	13.4	6.12	13.7	6.13	19	1.23	20.4	6.11	21	4.58
sfacloc2_4_95	NaN	NaN	14.3	3.01	14.8	3.55	20.6	5.21	17.4	3.53	17.4	3.64
spring	0.846	0.281	0.846	0.0434	0.846	0.0897	1.57	2.07	1.02	0.183	1.58	0.109
st_e15	7.67	0.00633	7.67	0.0126	7.67	0.011	7.67	0.0807	7.67	0.0188	7.67	0.0175
st_e29	-0.943	0.0614	-0.943	0.0197	-0.943	0.0294	0	0.108	-0.905	0.0405	-0.905	0.031
st_e32	-1.43	0.471	-1.43	0.238	-18.4	0.24	NaN	NaN	-1.43	0.227	-1.43	0.202
st_e35	NaN	NaN	6.49e+04	4.42	6.49e+04	4.73	9.26e+04	1.1	NaN	NaN	NaN	NaN
st_e36	-246	0.0108	-129	0.0556	-129	0.0573	-244	0.689	-245	0.0459	-245	0.0409
st_e38	7.2e+03	0.0075	7.2e+03	0.016	7.2e+03	0.0277	NaN	NaN	7.2e+03	0.0185	7.2e+03	0.0178
st_e40	NaN	NaN	30.4	0.0687	30.4	0.208	30.8	0.174	NaN	NaN	NaN	NaN
super3t	NaN	NaN	-0.671	300	-0.643	300	NaN	NaN	NaN	NaN	NaN	NaN
supplychain1_020306	NaN	NaN	4.38e+05	0.0587	4.38e+05	0.432	4.38e+05	0.298	4.38e+05	0.88	4.38e+05	0.803
supplychain1_022020	NaN	NaN	1.78e+06	3.65	1.8e+06	122	1.82e+06	46.6	1.78e+06	122	1.78e+06	142
supplychain1_030510	NaN	NaN	8.6e+05	0.211	8.6e+05	4.58	1e+06	3.26	8.6e+05	8.29	8.6e+05	7.28

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
supplychainp1_053050	NaN	NaN	4.08e+06	300	NaN	NaN	3.91e+06	136	NaN	NaN	NaN	NaN
supplychainr1_020306	NaN	NaN	4.38e+05	0.0376	4.38e+05	0.223	4.38e+05	0.675	5.94e+05	1.43	7.44e+05	0.903
supplychainr1_022020	NaN	NaN	1.78e+06	1.11	1.78e+06	6.31	1.82e+06	8.99	2.46e+06	300	5.21e+06	237
supplychainr1_030510	NaN	NaN	8.6e+05	0.05	8.6e+05	0.0999	8.6e+05	1.07	8.87e+05	2.44	8.87e+05	2.92
supplychainr1_053050	NaN	NaN	3.73e+06	83.5	NaN	NaN	3.82e+06	207	NaN	NaN	NaN	NaN
synheat	1.55e+05	0.568	1.68e+05	0.556	1.94e+05	0.559	1.68e+05	0.613	1.55e+05	0.545	1.56e+05	0.262
tanksize	1.27	0.568	1.27	0.212	1.27	0.215	NaN	NaN	1.27	0.203	1.27	0.164
transswitch0009p	5.3e+03	0.101	5.3e+03	300	5.3e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0009r	NaN	NaN	5.3e+03	300	5.3e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0014p	8.08e+03	0.127	NaN	NaN	NaN	NaN	NaN	NaN	8.08e+03	0.518	8.08e+03	0.48
transswitch0014r	NaN	NaN	NaN	NaN	8.08e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0030p	574	0.645	NaN	NaN	NaN	NaN	NaN	NaN	574	2.38	574	2.29
transswitch0030r	NaN	NaN	NaN	NaN	579	300	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0039p	4.19e+04	0.395	2	2.98	4.19e+04	3	NaN	NaN	4.19e+04	2.97	4.19e+04	3.7
transswitch0039r	NaN	NaN	2	300	4.19e+04	300	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0057p	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0057r	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0118p	1.29e+05	16.9	NaN	NaN	NaN	NaN	NaN	NaN	1.29e+05	21.5	1.29e+05	24.6
transswitch0118r	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0300p	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
transswitch0300r	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
transswitch2383wpp	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
transswitch2383wpr	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
transswitch2736spp	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
transswitch2736spr	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
tspn05	191	0.024	211	0.0744	194	0.064	191	0.122	191	0.054	191	0.0508
tspn08	291	0.0158	309	0.553	341	0.536	291	0.237	291	0.52	291	0.509
tspn10	225	0.0276	379	0.151	295	0.141	225	0.379	235	0.129	235	0.127
tspn12	266	0.046	NaN	NaN	460	0.34	266	0.456	266	0.327	266	0.151
tspn15	NaN	NaN	359	2.71	341	2.72	327	0.825	329	2.69	329	1.63
unitcommit2	5.9e+05	300	5.82e+05	0.188	5.8e+05	2.82	5.86e+05	300	5.85e+05	300	5.81e+05	300
uselinear	NaN	NaN	2.94e+03	300	2.94e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
var_con10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	444	11.4	444	14.4
var_con5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	278	20.2	278	16.7
wager	NaN	NaN	2.03e+04	0.328	2.03e+04	300	NaN	NaN	NaN	NaN	NaN	NaN
wastepaper3	NaN	NaN	0.0189	1.15	0.0272	1.17	0.14	3.12	0.649	1.15	NaN	NaN
wastepaper4	NaN	NaN	0.0113	1.47	0.0229	1.47	0.00415	104	0.187	1.46	NaN	NaN
wastepaper5	NaN	NaN	0.0288	2.57	0.0212	2.57	0.00565	14.2	0.187	2.56	NaN	NaN
wastepaper6	NaN	NaN	0.00759	7.1	0.00314	7.1	0.00961	181	0.0464	7.09	NaN	NaN
water3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
water4	NaN	NaN	907	300	927	300	NaN	NaN	973	300	973	28.8
waterful2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
waternd1	NaN	NaN	6.07e+05	0.523	6.4e+05	0.528	6.07e+05	8.94	6.92e+05	0.513	6.92e+05	0.353
waternd2	NaN	NaN	1.28e+06	2.1	3.2e+05	2.1	1.06e+06	31.6	1.06e+06	2.08	1.06e+06	1.88
waterno2_01	NaN	NaN	19.5	0.0382	19.5	0.486	29.1	1.24	133	0.497	133	0.367

Table 8 continued

Problem	Bonmin		Gurobi		SCIP		SHOT		S-B-MIQP		S-B-MIQP-ee	
	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time	Objective	Wall time
waterno2_02	NaN	NaN	39.6	0.379	55.3	2.35	104	2.36	266	4.31	272	3.82
waterno2_03	NaN	NaN	115	9.19	117	9.19	138	5.07	399	9.17	402	4.06
waterno2_04	NaN	NaN	145	25.1	187	25.1	NaN	NaN	651	25.1	NaN	NaN
waterno2_06	NaN	NaN	285	249	347	250	NaN	NaN	376	249	NaN	NaN
waterno2_09	NaN	NaN	953	300	1.06e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
waterno2_12	NaN	NaN	2.32e+03	300	2.48e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
waterno2_18	NaN	NaN	5.49e+03	300	5.72e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
waterno2_24	NaN	NaN	7.67e+03	300	7.86e+03	300	NaN	NaN	NaN	NaN	NaN	NaN
waters	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
watersbp	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
watersym1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
watersym2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
watertreatnd_conc	NaN	NaN	3.48e+05	0.202	3.48e+05	6.74	NaN	NaN	NaN	NaN	NaN	NaN
watertreatnd_flow	NaN	NaN	3.48e+05	7.11	3.48e+05	49	4.07e+05	111	NaN	NaN	NaN	NaN
waterx	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	968	0.247	968	0.318
waterz	NaN	NaN	911	300	940	300	NaN	NaN	NaN	NaN	NaN	NaN
windfac	NaN	NaN	0.254	0.0327	0.254	0.0383	NaN	NaN	0.75	300	0.75	0.0624

Acknowledgements A preliminary version of the algorithm proposed in this work has been presented in the Oberwolfach meeting “Mixed-integer Nonlinear Optimization: A Hatchery for Modern Mathematics” in August 2023 [23]. We thank Adrian Bürger, Clemens Zeile, Léo Simpson, and Ramin Abbasi-Esfeden for inspiring discussions.

Author Contributions **Andrea Ghezzi**: Writing - original draft, Writing - review & editing, Conceptualization, Visualization, Software. **Wim Van Roy**: Writing - original draft, Writing - review & editing, Conceptualization, Software. **Sebastian Sager**: Writing - review & editing, Supervision, Conceptualization. **Moritz Diehl**: Writing - review & editing, Supervision, Conceptualization, Resources, Project administration, Funding acquisition.

Funding Open Access funding enabled and organized by Projekt DEAL. Andrea Ghezzi and Moritz Diehl acknowledge funding from the European Union via ELO-X 953348, from DFG via projects 504452366 (SPP 2364), 560056112 (robust MPC), 535860958 (ALeSCo) and 525018088 (MAWERO), and from BMWK via 03EN3054B.

Wim Van Roy is supported by the Baekeland scholarship (Grant 182076), given by Atlas Copco Airpower NV, Wilrijk, Belgium, and by the Institute for the Promotion of Innovation through Science and Technology in Flanders (VLAIO), Belgium.

Sebastian Sager acknowledges funding from DFG via RTG 2297 and SPP 2331.

Data Availability All data analyzed during this study are publicly available, URLs are included in this article.

Code availability The full code is open-source, and specific references are included in this article. The packages we rely on are either open source or available for academic use.

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

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/>.

References

1. Abbasi-Esfeden, R., Roy, V., Swevers, W., J.: Iterative switching time optimization for mixed-integer optimal control problems. In: Proceedings of the European Control Conference (ECC), pp. 1–6. IEEE (2023)
2. Abhishek, K., Leyffer, S., Linderoth, J.T.: Filmint: An outer-approximation-based solver for nonlinear mixed integer programs, Mathematics and Computer Science Division, Argonne National Laboratory (2006). (Preprint ANL/MCS-P1374-0906)
3. Achterberg, T.: SCIP: solving constraint integer programs. *Math. Program. Comput.* **1**, 1–41 (2009)
4. Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi - a software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **11**(1), 1–36 (2019)
5. Axelsson, H., Wardi, Y., Egerstedt, M., Verriest, E.: Gradient descent approach to optimal mode scheduling in hybrid dynamical systems. *J. Optim. Theory Appl.* **136**(2), 167–186 (2008)
6. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software* **24**(4–5), 597–634 (2009)

7. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. *Acta Numer* **22**, 1–131 (2013)
8. Bertacco, L., Fischetti, M., Lodi, A.: A feasibility pump heuristic for general mixed-integer problems. *Discret. Optim.* **4**(1), 63–76 (2007)
9. Bestuzheva, K., Chmiela, A., Müller, B., Serrano, F., Vigerske, S., Wegscheider, F.: Global optimization of mixed-integer nonlinear programs with SCIP 8. *J. Global Optim.*, 1–24 (2023)
10. Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. In: *Proceedings of the IFAC World Congress*, pp. 242–247. Pergamon Press (1984)
11. Bonami, P., Biegler, L., Conn, A., Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: *An Algorithmic Framework For Convex Mixed Integer Nonlinear Programs*. Tech. rep., IBM T. J. Watson Research Center (2005)
12. Boyd, S., Vandenberghe, L.: *Convex Optimization*. University Press, Cambridge (2004)
13. Bürger, A., Zeile, C., Altmann-Dieses, A., Sager, S., Diehl, M.: Design, implementation and simulation of an MPC algorithm for switched nonlinear systems under combinatorial constraints. *J. Process Control* **81**, 15–30 (2019)
14. Bürger, A., Bohlayer, M., Hoffmann, S., Altmann-Dieses, A., Braun, M., Diehl, M.: A whole-year simulation study on nonlinear mixed-integer model predictive control for a thermal energy supply system with multi-use components. *Appl. Energy* **258**, 114064 (2020)
15. Bürger, A., Zeile, C., Hahn, M., Altmann-Dieses, A., Sager, S., Diehl, M.: pycombina: An open-source tool for solving combinatorial approximation problems arising in mixed-integer optimal control. *Proceedings of the IFAC World Congress* **53**, 6502–6508 (2020)
16. Bürger, A., Bull, D., Sawant, P., Bohlayer, M., Klotz, A., Beschütz, D., Altmann-Dieses, A., Braun, M., Diehl, M.: Experimental operation of a solar-driven climate system with thermal energy storages using mixed-integer nonlinear model predictive control. *Optimal Control Applications and Methods*, 1–27 (2021)
17. Bürger, A., Zeile, C., Altmann-Dieses, A., Sager, S., Diehl, M.: A Gauss-Newton-based decomposition algorithm for nonlinear mixed-integer optimal control problems. *Automatica* **152**, 110967 (2023)
18. Büskens, C., Wassel, D.: The ESA NLP solver WORHP. *Modeling and optimization in space engineering*, pp. 85–110. (2013)
19. Bussieck, M.R., Drud, A.S., Meeraus, A.: Minplib - a collection of test models for mixed-integer nonlinear programming. *INFORMS J. Comput.* **15**(1), 114–119 (2003)
20. Bynum, M.L., Hackebeil, G.A., Hart, W.E., Laird, C.D., Nicholson, B.L., Sirola, J.D., Watson, J.P., Woodruff, D.L.: *Pyomo—optimization modeling in Python*, vol 67, 3rd edn. Springer Science & Business Media (2021)
21. Byrd, R.H., Nocedal, J., Waltz, R.A.: KNITRO: An integrated package for nonlinear optimization. In: Roma, M. (ed.) *Pillo G*, pp. 35–59. *Large Scale Nonlinear Optimization*, Springer Verlag (2006)
22. Dakin, R.: A tree-search algorithm for mixed integer programming problems. *Comput. J.* **8**, 250–255 (1965)
23. Diehl, M., Ghezzi, A., Van Roy, W., Sager, S.: A sequential mixed-integer quadratic programming algorithm for solving MINLP arising in optimal control. *Mixed-integer Nonlinear Optimization: A Hatchery for Modern Mathematics* (35), (2023) <https://doi.org/10.4171/OWR/2023/35>
24. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**, 201–213 (2002)
25. Duran, M., Grossmann, I.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.* **36**(3), 307–339 (1986)
26. Exler, O., Schittkowski, K.: A trust region sqp algorithm for mixed-integer nonlinear programming. *Optimization Letters* **1**, 269–280 (2007)
27. Fiacco, A.: *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press, New York (1983)
28. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. *Math. Program.* **66**, 327–349 (1994)
29. Forrest, J., Lougee-Heimer, R.: Cbc user guide. In: *Emerging theory, methods, and applications*, pp. 257–277. *INFORMS* (2005)
30. Garey, M., Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, W.H (1979)
31. Geoffrion, A.: Generalized Benders Decomposition. *J. Optim. Theory Appl.* **10**, 237–260 (1972)

32. Ghezzi, A., Simpson, L., Buerger, A., Zeile, C., Sager, S., Diehl, M.: A Voronoi-based mixed-integer Gauss-Newton algorithm for MINLP arising in optimal control. In: Proceedings of the European Control Conference (ECC), (2023)
33. Gupta, O., Ravindran, A.: Branch and Bound experiments in convex nonlinear integer programming. *Manage. Sci.* **31**, 1533–1546 (1985)
34. Gurobi Optimization, LLC (2024) Gurobi Optimizer Reference Manual. <https://www.gurobi.com>, Last accessed: 2024-04-01
35. Hart, W.E., Watson, J.P., Woodruff, D.L.: Pyomo: modeling and solving mathematical programs in Python. *Math. Program. Comput.* **3**(3), 219–260 (2011)
36. HSL (2024) A collection of Fortran codes for large scale scientific computation. <http://www.hsl.rl.ac.uk>, Last accessed: 2024-04-01
37. Huangfu, Q., Hall, J.J.: Parallelizing the dual revised simplex method. *Math. Program. Comput.* **10**(1), 119–142 (2018)
38. IBM Corp (2022) IBM ILOG CPLEX V22.1, User's Manual for CPLEX. <https://www.ibm.com/products/ilog-cplex-optimization-studio>, Last accessed: 2024-04-01
39. Köppe, M.: On the complexity of nonlinear mixed-integer optimization. In: Mixed Integer Nonlinear Programming, pp. 533–557. Springer (2011)
40. Kronqvist, J., Lundell, A., Westerlund, T.: The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *J. Global Optim.* **64**, 249–272 (2016)
41. Kronqvist, J., Bernal, D.E., Grossmann, I.E.: Using regularization and second order information in outer approximation for convex MINLP. *Math. Program.* **180**(1), 285–310 (2020)
42. Lee, H.J., Teo, K.L., Rehbock, V., Jennings, L.S.: Control parametrization enhancing technique for optimal discrete-valued control problems. *Automatica* **35**(8), 1401–1407 (1999)
43. Li, D., Sun, X.: Nonlinear integer programming, vol. 84. Springer (2006)
44. Li, X., Chen, Y., Barton, P.I.: Nonconvex generalized benders decomposition with piecewise convex relaxations for global optimization of integrated process design and operation problems. *Industrial & engineering chemistry research* **51**(21), 7287–7299 (2012)
45. Lundell, A., Kronqvist, J.: Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in SHOT. *J. Global Optim.* **82**(4), 863–896 (2022)
46. Lundell, A., Kronqvist, J., Westerlund, T.: The supporting hyperplane optimization toolkit for convex MINLP. *J. Global Optim.* **84**(1), 1–41 (2022)
47. Matthias M.: Visualizations of Mittelman benchmarks. <https://mattmilten.github.io/mittelman-plots/>, Last accessed: 2024-04-01 (2024)
48. Messerer, F., Baumgärtner, K., Diehl, M.: Survey of sequential convex programming and generalized Gauss-Newton methods. *ESAIM: Proceedings and Surveys* **71**, 64–88 (2021)
49. MINLPLib (2025) A library of mixed-integer and continuous nonlinear programming instances. <https://www.minlplib.org/instances.html>, Last updated: 2025-05-08, Git hash: 2f1d9c1a
50. Misener, R., Floudas, C.A.: Antigone: algorithms for continuous/integer global optimization of nonlinear equations. *J. Global Optim.* **59**(2–3), 503–526 (2014)
51. MOSEK ApS (2024) <https://www.mosek.com>, Last accessed: 2024-04-01
52. Quesada, I., Grossmann, I.: An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Comput. Chem. Eng.* **16**, 937–947 (1992)
53. Rawlings, J.B., Mayne, D.Q., Diehl, M.M.: Model Predictive Control: Theory, Computation, and Design, 2nd edn. Nob Hill (2017)
54. Robuschi, N., Zeile, C., Sager, S., Braghin, F.: Multiphase mixed-integer nonlinear optimal control of hybrid electric vehicles. *Automatica* **123**, 109325 (2021)
55. Sager, S., Jung, M., Kirches, C.: Combinatorial integral approximation. *Math. Methods Oper. Res.* **73**(3), 363–380 (2011)
56. Sahinidis, N.V.: Baron: A general purpose global optimization software package. *J. Global Optim.* **8**, 201–205 (1996)
57. Sahinidis, N.V.: Mixed-integer nonlinear programming 2018. *Optim. Eng.* **20**, 301–306 (2019)
58. Smith, E.M., Pantelides, C.C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex minlps. *Computers & Chemical Engineering* **23**(4–5), 457–478 (1999)
59. Thorsteinsson, E.S.: Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In: Principles and Practice of Constraint Programming—CP 2001: 7th International Conference, Springer, pp 16–30 (2001)

60. Tsang, T., Himmelblau, D., Edgar, T.: Optimal control via collocation and non-linear programming. *International Journal on Control* **21**, 763–768 (1975)
61. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **106**(1), 25–57 (2006)
62. Westerlund, T., Pettersson, F.: A cutting plane method for solving convex MINLP problems. *Comput. Chem. Eng.* **19**, S131–S136 (1995)
63. Zeile, C., Robuschi, N., Sager, S.: Mixed-integer optimal control under minimum dwell time constraints. *Math. Program.* **188**(2), 653–694 (2021)

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