informs
https://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING

Vol. 00, No. 0, Xxxxx 0000, pp. 000–000

Submitted to *INFORMS Journal on Computing*

ISSN 0899-1499, EISSN 1526-5528

Enhanced Outer-Approximation Methods for MINLP via Convexification and Bound Tightening

Zedong Peng, Kaiyu Cao
Davidson School of Chemical Engineering, Purdue University, peng372@purdue.edu, cao425@purdue.edu

Kevin C. Furman
ExxonMobil Upstream Research Co, kevin.c.furman@exxonmobil.com

Can Li
Davidson School of Chemical Engineering, Purdue University, canli@purdue.edu

Ignacio E. Grossmann
Department of Chemical Engineering, Carnegie Mellon University, grossmann@cmu.edu

David E. Bernal Neira
Davidson School of Chemical Engineering, Purdue University, dbernaln@purdue.edu

Abstract. Advancements in convexification and domain-reduction techniques have significantly enhanced the performance of global optimization solvers for challenging Mixed-Integer Nonlinear Programming (MINLP) problems. However, the great potential of these techniques has not yet been fully harnessed in decomposition-based MINLP solvers. Motivated by this consideration, we propose improving the performance of the Outer-Approximation (OA) method by integrating several convexification cuts and Bound Tightening (BT) techniques. To effectively solve convex MINLP problems, both multi-tree and single-tree (i.e., known as the LP/NLP Branch-and-Bound (B&B) method) settings of the OA method are investigated. Moreover, the recently developed global OA and global LP/NLP B&B methods are extended to include these techniques for non-convex MINLP problems. These new methods have been developed and incorporated into the open-source **M**ixed-**I**nteger **N**onlinear **D**ecomposition **T**oolbox for **Py**omo-MindtPy. Extensive computational experiments were conducted using the MINLP benchmark library MINLPLib to validate the effectiveness and reliability of our proposed methods. The results show that incorporating the convexification and domain reduction techniques into the (global) OA and (global) LP/NLP B&B methods significantly reduces the computational time and the number of iterations required for the solution.

Key words: Mixed-Integer Nonlinear Programming, Outer-Approximation, LP/NLP based Branch and Bound, Domain Reduction

1. Introduction

Mixed-integer nonlinear programming (MINLP) is a general and powerful modeling framework designed to optimize both discrete and continuous variables subject to linear and nonlinear constraints. Thus, a wide variety of problems are effectively represented using this framework, such as portfolio optimization (Bonami and Lejeune 2009), unit commitment (Bacci et al. 2024), network design (Bragalli et al. 2012), process synthesis (Kocis and Grossmann 1988), and production scheduling (Sahinidis and Grossmann 1991). However, solving these problems is computationally challenging due to (1) the combinatorial complexity introduced by integer variables, (2) the nonlinearity present in objective functions and/or constraints, and (3) the possible non-convexity arising from many practical applications.

Generally, MINLP problems are non-convex by definition due to the inclusion of integer variables and possible non-convex functions in the objective function and constraints. However, an MINLP problem is often classified as convex if its continuous relaxation is convex. Algorithms for convex MINLP problems fall primarily into two categories: Branch-and-Bound (B&B)-based methods, including B&B (Dakin 1965), Branch-and-Cut (B&C) (Vigerske and Gleixner 2018), and Branch-and-Reduce (B&R) (Ryoo and Sahinidis 1996); and decomposition-based methods, such as the Outer Approximation (OA) (Duran and Grossmann 1986), Extended Cutting Plane (ECP) (Westerlund and Pettersson 1995), Extended Supporting Hyperplane (ESH) (Kronqvist et al. 2016), Feasibility Pump (FP) (Bonami et al. 2009, Bernal et al. 2020), and Generalized Benders Decomposition (GBD) (Geoffrion 1972). The success of these methods is evident from the numerous software implementations available, including BARON (Sahinidis 1996), SCIP (Bestuzheva et al. 2023), ANTIGONE (Misener and Floudas 2014), Couenne (Belotti et al. 2009), SHOT (Lundell et al. 2022), BONMIN (Bonami et al. 2008), DICOPT (Kocis and Grossmann 1989), MindtPy (Bernal et al. 2018) and Pajarito (Coey et al. 2020).

Among decomposition methods, the OA method is widely recognized as one of the most reliable and effective approaches for solving convex MINLP problems. By alternately solving tractable MILP master problems and nonlinear programming (NLP) subproblems, the OA method iteratively generates linear inequality constraints to approximate nonlinear constraints using first-order Taylor extensions. To avoid solving multiple similar MILP master problems that differ only in their accumulated OA cuts, the Linear Programming and Nonlinear Programming-based B&B (LP/NLP-B&B) (Quesada and Grossmann 1992) was proposed, also known as the ”single-tree” method. This method constructs a single B&B tree, where the MILP problem is dynamically updated with the newly generated OA cuts, and the NLP subproblems are solved at the integer nodes of the tree. Besides this, several extensions to OA methods have been proposed, such as quadratic cut OA (Su et al. 2018), conic-based OA (Coey et al. 2020), regularized OA (Bernal et al. 2022), and decomposition-based OA (Muts et al. 2020).

Over the decades, B&B-based solvers have significantly improved their performance. They benefit from the extensive preprocessing, cutting plane, and heuristic techniques developed for MINLP and inherited from mixed-integer linear programming (MILP) solvers. However, many of these techniques have not yet been fully leveraged in decomposition-based solvers. While the LP/NLP-based B&B method can take advantage of the techniques implemented in MILP solvers, to our knowledge, the effect of the techniques specialized for MINLP has not been tested in decomposition-based solvers. Moreover, a significant challenge for decomposition methods, including the OA method, lies in their initialization. Typically, these methods start with a poor approximation with all nonlinearities relaxed and refine it by accumulating linear constraints. As a result, early iterations are spent merely obtaining a reasonable approximation of the nonlinear constraints, leading to high computational costs. For the OA method, the poor approximation may result in infeasible NLP subproblems in early iterations, and no primal bound can be obtained. Additionally, the OA cuts generated from these infeasible NLP subproblems are generally not as strong as those derived from the optimal NLP solutions. In this work, we aim to address this limitation by incorporating domain-reduction and convexification techniques, strategies proven effective in B&B solvers, to tighten the initial approximation and reduce computational overhead.

Domain reduction techniques can be classified according to the arguments used. The Feasibility-based Bound Tightening (FBBT) (Belotti et al. 2012) techniques typically rely on interval arithmetic and exploit the relationships between the constraints and variable bounds (i.e., through forward and backpropagation). This iterative process may converge only in the limit; thus, a termination criterion is usually set based on either the number of iterations or a threshold for improvement. As a result, infeasible subdomains of the search space can be eliminated while all feasible solutions are preserved. On the other hand, optimality-based bound tightening (OBBT) (Gleixner et al. 2017, Zhang et al. 2020) techniques typically require solving convex relaxations of the original problem to compute the tightest bounds of variables. For instance, the standard OBBT technique tightens the bounds by solving auxiliary optimization problems for each variable. These problems minimize or maximize each single variable, considering all problem constraints and a possible objective cut-off. Since the objective cut-off is included, the OBBT method further eliminates feasible points from the search space while ensuring that at least one globally optimal solution is preserved. Additionally, Marginals-based bound tightening (MBBT) (Puranik and Sahinidis 2017) uses the optimality gap and dual variable information to derive inequalities that will only cut off feasible points with a worse objective than the best-known objective. A comprehensive review of common domain reduction techniques and their implementation on B&B algorithms can be found in Puranik and Sahinidis (2017).

For non-convex MINLP problems, the first-order Taylor expansion of nonlinear constraints can no longer guarantee a valid underestimator to the original feasible region. To address this challenge, global MINLP solvers, such as BARON (Sahinidis 1996), Couenne (Belotti et al. 2009), and SCIP (Achterberg 2009)) typically implement spatial B&B (sBB) algorithms (Smith and Pantelides 1999), in conjunction with convexification techniques to construct valid convex and concave relaxations of non-convex functions. Linearization of convex and concave relaxations can be further generated. This idea can be extended to the OA method to generate valid cutting planes for non-convex constraints and tighten the initial approximation. Effective convexification techniques include piecewise linear relaxation (Floudas 2013), αBB and γBB (Adjiman and Floudas 1996, Akrotirianakis and Floudas 2004, Maranas and Floudas 1992), the auxiliary variable method (AVM) (Smith and Pantelides 1997, Tawarmalani and Sahinidis 2013), McCormick relaxation (McCormick 1976, Tsoukalas and Mitsos 2014) and semidefinite relaxation (SDR) (Tawarmalani and Sahinidis 2001, Luo et al. 2010). Among these, both the AVM and McCormick relaxations are generally applicable to non-convex factorable functions. The AVM introduces an auxiliary variable and a corresponding equality constraint for each intermediate nonlinear factor in the non-convex function. The convex and concave envelopes obtained from each component can be further linearized using the sandwich algorithm (Tawarmalani and Sahinidis 2013). In contrast, the McCormick relaxations use a recursive approach to propagate the relaxations of univariate intrinsic functions, which maintains the dimension of the original function (Chachuat 2013). Since the relaxations are generally non-smooth, subgradient propagation is used to determine the valid affine under- and overestimators for nonlinear relaxations (Smith and Pantelides 1997). Although AVM incorporates many auxiliary variables and constraints that can lead to increased computational time, its relaxations have been reported to be at least as tight and, in many cases, tighter than McCormick relaxations due to repeated terms (Tsoukalas and Mitsos 2014).

Motivated by these considerations, this paper aims to investigate the efficacy of convexification and domain reduction techniques within the OA and LP/NLP B&B methods, applicable to both convex and non-convex MINLP problems. More specifically, we want to answer three questions.

1. Whether bound tightening and convexification techniques can accelerate the convergence of the OA and LP/NLP B&B methods.
2. Can these two techniques reduce the number of infeasible NLP subproblems that generate weak polyhedral outer approximations and usually appear in early iterations?
3. If so, is it worth applying these two techniques in terms of solution time?

This work builds upon and draws inspiration from early investigations by Wei et al. (2005), Furman et al. (2005), Furman (2006). These works explored the use of these methods in both deterministic and stochastic settings for convex and non-convex MINLP problems, serving as early investigations of the ideas, though not comprehensively tested. It is understood that domain reduction techniques can be used during the presolve stage and at each node within the B&B tree, known as the branch-and-reduce method. Similarly, in the OA method, these techniques can be employed in both the pre-solve phase and during solving integer-fixed NLP subproblems. However, this work focuses specifically on the impact of domain-reduction methods at the method's initialization stage.

The remainder of the paper is organized as follows: Section 2 provides a concise overview of the OA, LP/NLP B&B, global OA and global LP/NLP B&B, convexification method, and the BT technique. In Section 3, we introduce our proposed convexification and bound-tightening-based (global) OA and (global) LP/NLP B&B methods. The implementation of the proposed methods and the computational results are detailed and presented in Section 4. Finally, conclusions are drawn in Section 5, and some future research directions are identified.

2. Preliminaries

The general form of a MINLP problem can be written as follows.

$$\begin{aligned} \min_{x,y} \quad & f(x,y) \\ s.t. \quad & g_j(x,y) \le 0, \forall j = 1,\dots,l \\ & Ax + By \le b, \\ & x \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n, \\ & y \in \left\{\underline{y}, \dots, \overline{y}\right\} \subseteq \mathbb{Z}^m, \end{aligned} \tag{MINLP}$$

where x and y represent continuous and discrete variables, respectively. Their respective upper and lower bounds are determined by over- and underbars. The objective function $f(x,y)$ and constraints $g_j(x, y)$ are potentially nonlinear. Throughout this paper, the nonlinear functions in both the objective function and constraints are assumed to be differentiable, and no restrictions on their convexity are set. Thus, if any of the nonlinear functions in problem (MINLP) is non-convex or if there exists any nonlinear equality constraint, the problem should be regarded as a non-convex MINLP. Note that the (nonlinear) objective function can always be transformed to a linear one using the epigraph formulation (i.e., $f(x, y) \leq \mu$ where the continuous variable μ represents the objective value).

2.1. (Global) Outer Approximation

The OA method was initially proposed to solve convex MINLP problems with guaranteed globality (Duran and Grossmann 1986). It generates an iteratively improving polyhedral approximation of problem (MINLP), where the nonlinear feasible region is outer approximated by cutting planes. To guarantee global convergence in convex MINLP problems, it relies on the following assumptions: (1) the nonlinear objective function and constraints $f, g_1, \ldots, g_l : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are convex and continuously differentiable; (2) the linear constraints $Ax + By \leq b$ define a bounded polyhedron (i.e., all variables must be bounded); and (3) for each feasible integer combination y, a constraint qualification holds (e.g., Slater's condition (Slater 2013)). Herein, we briefly introduce the main steps of the OA method. For more details, the reader may refer to Duran and Grossmann (1986).

The OA method constructs a polyhedral approximation of the nonlinear feasible region to determine the values of the integer variables. Once an integer combination is obtained, it is fixed, and the corresponding continuous variables are determined by solving a continuous optimization problem. Specifically, the solutions (x^k, y^k) obtained in the previous iteration k are used to construct a polyhedral outer approximation of problem (MINLP), where first-order Taylor series expansions underestimate the nonlinear functions. This results in the following MILP subproblem (i.e., problem (OA-master)), which is solved to obtain the following integer combination y^{k+1}. Note that I_k is the index set containing all the nonlinear constraints active at solution (x^k, y^k).

$$
\begin{aligned}
\min_{x,y,\mu} \quad & \mu \\
s.t. \quad & f\left(x^i, y^i\right) + \nabla f\left(x^i, y^i\right)^{\mathrm{T}} \begin{bmatrix} x - x^i \\ y - y^i \end{bmatrix} \leq \mu \quad \forall i = 1, \ldots, k \\
& g_j\left(x^i, y^i\right) + \nabla g_j\left(x^i, y^i\right)^{\mathrm{T}} \begin{bmatrix} x - x^i \\ y - y^i \end{bmatrix} \leq 0 \quad \forall i = 1, \ldots, k, \forall j \in I_i \\
& Ax + By \leq b, \\
& x \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n, \\
& y \in \left\{\underline{y}, \ldots, \overline{y}\right\} \subseteq \mathbb{Z}^m, \\
& \mu \in \mathbb{R}
\end{aligned}
\tag{OA-master}
$$

Due to the assumption of convexity, the polyhedral outer approximation overestimates the original nonlinear feasible region; thus, the optimal solution of the (OA-master) problem provides a valid lower bound to the (MINLP) problem (i.e., LB^{k+1}). Notably, in the first iteration, a continuous relaxation of the (MINLP) problem is often solved to provide the initial LB (i.e., LB^0) and polyhedral approximation.

Once the new integer combination y^{k+1} is obtained, the following (NLP-I) subproblem can be constructed by fixing the values of the integer variables. This convex NLP subproblem is solved to determine the values of the corresponding continuous variables x^{k+1}.

$$
\begin{aligned}
\min_{x} \quad & f(x, y^{k+1}) \\
s.t. \quad & g_j\left(x, y^{k+1}\right) \leq 0 \quad \forall j = 1, \ldots, l \\
& Ax + By^{k+1} \leq b, \\
& x \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n,
\end{aligned}
\tag{NLP-I}
$$

If the (NLP-I) subproblem is feasible and solved to optimality, its optimum provides a valid upper bound to the (MINLP) problem (i.e., UB^{k+1}). It is worth noting that the NLP subproblem may be infeasible with the given integer values, particularly in early iterations. In this case, an alternative feasibility subproblem can be formulated to minimize the norm p of the constraint violations s (i.e., (NLP-f)). Instead, this subproblem is solved to determine the continuous variables x^{k+1}. However, since this solution (x^{k+1}, y^{k+1}) is not feasible for the problem (MINLP), no valid UB can be obtained in this iteration.

$$\begin{aligned} \min_{x,s} \quad & \|s\|_p \\ s.t. \quad & g_j\left(x, y^{k+1}\right) \leq s_j \quad \forall j = 1, \ldots, l \\ & Ax + By^{k+1} \leq b, \\ & x \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n, \end{aligned} \tag{NLP-f}$$

When the difference between the upper and lower bounds exceeds the predetermined optimality gap, the polyhedral outer approximation of the nonlinear feasible region is improved by iteratively adding new linear constraints to problem (OA-master), which are generated based on the complete solution (x^{k+1}, y^{k+1})) as follows (i.e., (OA-cut).

$$\begin{aligned} & f\left(x^{k+1}, y^{k+1}\right) + \nabla f\left(x^{k+1}, y^{k+1}\right)^{\mathrm{T}} \begin{bmatrix} x - x^{k+1} \\ y - y^{k+1} \end{bmatrix} \leq \mu \\ & g_j\left(x^{k+1}, y^{k+1}\right) + \nabla g_j\left(x^{k+1}, y^{k+1}\right)^{\mathrm{T}} \begin{bmatrix} x - x^{k+1} \\ y - y^{k+1} \end{bmatrix} \leq 0 \quad \forall j \in I_{k+1} \end{aligned} \tag{OA-cut}$$

These linear constraints are often referred to as OA cuts because, due to the convexity assumption, they can cut off the infeasible parts of the search space. Thus, adding new OA cuts can refine the polyhedral outer approximation and ensure that the previously obtained integer combination is not obtained in the subsequent iteration unless it is the optimal integer solution. Note that although the solutions from the feasibility subproblem (NLP-f) do not necessarily produce a valid UB, they are still used to generate new OA cuts. Thus, the key of the OA method is the progressive accumulation of OA cuts, which incrementally narrows the gap between the best LB and UB. Iteratively solving the (OA-master), (NLP-I), (NLP-f) problems continues until the LB and UB converge because either a new integer combination should be obtained or the optimality should be verified after each iteration. The detailed steps of the OA algorithm are summarized in Algorithm 1.

As mentioned in the introduction section, when the OA method is applied to non-convex MINLP problems, it cannot be guaranteed to generate a valid polyhedral outer approximation of the non-convex feasible region; in other words, the OA cuts generated by (OA-cut) may be ineffective and exclude feasible parts of the search space. Thus, the resulting (OA-master) problem cannot provide a valid LB and could even become infeasible while the original MINLP problem is feasible. To address this issue, a McCormick relaxation-based OA method, the Global Outer Approximation (GOA) method, was recently proposed to guarantee global optimality for non-convex MINLP problems. Its main idea is to use affine under- and overestimators for the convex and concave McCormick relaxations of the (MINLP) problem to construct a valid polyhedral outer approximation of the feasible region defined by the non-convex constraints.

The McCormick relaxation is a general method used to construct convex and concave relaxations of non-convex factorable functionsTawarmalani and Sahinidis (2013). It uses the developed McCormick propagation rules to relax sums, products, univariate compositions, and multivariate compositions in functions (Tsoukalas and Mitsos 2014). Since the relaxations often involve non-differentiable approximations of non-convex functions, they are generally non-smooth. This non-smoothness necessitates using subgradient propagation to construct valid affine approximations at selected linearization points (i.e., in both smooth and non-smooth regions). For instance, with a given continuous non-convex function $f : X \to \mathbb{R}$, let (1) $f^{cv}, f^{cc} : X \to \mathbb{R}$ be a convex and concave relaxation of f on X, respectively and (2) $s_f^{cv}(\bar{x}), s_f^{cc}(\bar{x})$ are a convex and concave subgradient of f^{cv}, f^{cc} at $\bar{x} \in X$, respectively. Then, the affine relaxations of the function f can be derived as in (MC-relaxation). The detailed rules for the calculation of McCormick relaxations and subgradients can be found in (Mitsos et al. 2009, Tsoukalas and Mitsos 2014).

$$\begin{aligned} f^{cv}(\bar{x}) + (s^{cv}(\bar{x}))^{\mathrm{T}}(x - \bar{x}) \leq f(x), \quad \forall x \in X \\ f^{cc}(\bar{x}) + (s^{cc}(\bar{x}))^{\mathrm{T}}(x - \bar{x}) \geq f(x), \quad \forall x \in X \end{aligned} \tag{MC-relaxation}$$

Similar to the procedure of the OA method, the GOA method iteratively solves a sequence of relaxed MILP master problems and integer-fixed NLP subproblems to determine the complete solution of the (MINLP) problem. The major difference is that, in each iteration, the relaxed MILP master problem is constructed by using the McCormick relaxation-based cuts as follows.

$$
\begin{aligned}
&f^{cv}\left(x^k, y^k\right) + (s_f^{cv}\left(x^k, y^k\right))^{\mathrm{T}} \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le \mu \\
&g_j^{cv}\left(x^k, y^k\right) + (s_{g_j}^{cv}\left(x^k, y^k\right))^{\mathrm{T}} \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le 0 \quad \forall j \in I_k
\end{aligned}
\tag{MC-cut}
$$

It is worth noting that these McCormick relaxation-based cuts cannot necessarily cut off all the infeasible parts of the nonlinear feasible region. Thus, it is impossible to construct an MILP master problem that is equivalent to the original MINLP problem; in other words, when the integer values satisfy the (GOA-master) problem but are not feasible to the (MINLP) problem, the algorithm will repeat cycling through this integer combination. To guarantee the algorithm's convergence, no-good (integer) cuts are introduced to exclude the previously explored integer combinations from future iterations, as shown in (NoGood-cut). Alternatively, a Tabu list can also be used to exclude the explored integer combinations.

$$
\sum_{j \in V_1} y_j - \sum_{j \in V_0} y_j \le |V_1| - 1 \tag{NoGood-cut}
$$

where $V_1 = \{j | y_j = 1|\}$ and $V_0 = \{j | y_j = 0|\}$.

With these, the (OA-master) problem is reformulated as the (GOA-master) problem as follows. While the NLP subproblems should remain the same as the ones in the OA method (i.e., (NLP-I) and (NLP-f) subproblems).

$$
\begin{aligned}
\min_{x,y,\mu} \quad & \mu \\
s.t. \quad & f^{cv}\left(x^i, y^i\right) + \left(s_f^{cv}\left(x^i, y^i\right)\right)^{\mathrm{T}} \begin{bmatrix} x - x^i \\ y - y^i \end{bmatrix} \leq \mu \quad \forall i = 1, \ldots, k \\
& g_j^{cv}\left(x^i, y^i\right) + \left(s_{g_j}^{cv}\left(x^i, y^i\right)\right)^{\mathrm{T}} \begin{bmatrix} x - x^i \\ y - y^i \end{bmatrix} \leq 0 \quad \forall i = 1, \ldots, k, \forall j \in I_i \\
& Ax + By \leq b, \\
& \sum_{j \in V_1^i} y_j - \sum_{j \in V_0^i} y_j \leq |V_1^i| - 1 \quad \forall i = 1, \ldots, k \\
& x \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n, \\
& y \in \left\{\underline{y}, \ldots, \overline{y}\right\} \subseteq \mathbb{Z}^m, \\
& \mu \in \mathbb{R}
\end{aligned}
\tag{GOA-master}
$$

By using either the no-good cuts or the Tabu list, it ensures that the integer combinations from the previous iterations $\hat{y} \in \left\{y^i \mid i = 1, \ldots, k-1\right\}$ are infeasible to the (GOA-master) problem in the current iteration k. Thus, a new integer combination y^k can always be obtained. Given that there is a finite number of integer combinations in the search space of the (MINLP) problem, the algorithm can converge to the global optimum as long as the NLP subproblems are solved to global optimality (Kesavan et al. 2004). It is worth noting that using either the no-good cuts or the Tabu list may potentially cut off feasible points of the original MINLP problem, meaning that the optimum of the (GOA-master) problem cannot be taken as a valid LB. Thus, after the algorithm converges, an additional step is required to obtain the valid LB. In this step, the (GOA-master) problem must be solved again, but with only the (MC-cut) added.

Given the similarities between the OA and GOA methods, the detailed steps of the GOA algorithm are also summarized in Algorithm 1.

2.2. (Global) LP/NLP-Based Branch and Bound

As demonstrated in the OA method (i.e., Algorithm 1), a new (OA-master) problem must be constructed in every iteration and then solved to obtain an integer combination. This feature characterizes the OA method as a multi-tree method. However, it can be observed that (1) the

Algorithm 1 OA and GOA Algorithms

input: The (MINLP) problem and accepted optimality gap $\epsilon \geq 0$

Initialize: $k = 1$, $LB^0 = -\infty$, $UB^0 = +\infty$;

Solve the continuous relaxation of the (MINLP) problem to obtain an initial solution $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$;

Update $LB^0 = f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$, generate cutting planes at the initial solution $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ according to (cut[1]), and construct the (MILP-master[2]) problem;

while $UB^{k-1} - LB^{k-1} > \epsilon$ **do**

  Solve the (MILP-master[2]) problem to obtain $\mathbf{y^k}$ and LB^k;

  Solve the (NLP-I) subproblem with integer variables fixed as $\mathbf{y^k}$;

  if the (NLP-I) subproblem is feasible **then**

    Obtain $\mathbf{x^k}$ and update $UB^k = \min\left(f(\mathbf{x^k}, \mathbf{y^k}), UB^{k-1}\right)$;

  else

    Solve the (NLP-f) feasibility subproblem to obtain $\mathbf{x^k}$ and set $UB^k = UB^{k-1}$;

  end if

  Generate cutting planes at the solution $\mathbf{x^k}, \mathbf{y^k}$ according to (cut[1]) and add them to the (MILP-master[2]) problem;

  if GOA algorithm is selected **then**

    Generate no-good cuts at the solution $\mathbf{x^k}, \mathbf{y^k}$ according to (NoGood-cut) and add them to the (GOA-master) problem;

  end if

  $k = k + 1$;

end while

if GOA algorithm is selected **then**

  Solve the (GOA-master) problem without the no-good cuts to obtain the valid LB of the (MINLP) problem;

end if

return Best found solution

Note:[1]use (OA-cut) for OA and (MC-cut) for GOA. [2]use (OA-master) for OA and (GOA-master) for GOA.

only difference between the (OA-master) problem in the current iteration and the one in the previous iteration is the addition of some newly generated OA cuts, and (2) most of the total computational time is usually spent on solving the MILP (OA-master) problem through the B&B method. Motivated by these, Quesada and Grossmann (Quesada and Grossmann 1992) proposed a single-tree method (i.e., LP/NLP B&B method) that integrates the OA method within the B&B method to avoid solving too many similar MILP problems. Its main idea is to construct only a single B&B tree for the (OA-master) problem, which requires to be updated iteratively with new OA cuts. Herein, we briefly introduce the main steps of the LP/NLP B&B method.

Similar to the OA method, after obtaining the initial polyhedral outer approximation from the initialization step, the first (OA-master) problem is constructed, based on which the single B&B tree is built. Following the standard B&B procedure, a relaxed LP problem is solved in each node of the search tree, where the integer variables are relaxed as continuous ones and nonlinear constraints are approximated by using the OA cuts. Once a new incumbent integer solution $(\hat{x}, \hat{y})$ is found at a specific node, $\hat{y}$ is taken as a new integer combination y^k as in the OA method if it has not been explored. At this point, the best LB of the current B&B tree is considered a valid LB on the optimum of the original MINLP problem. Then, the (NLP-I) subproblem with the given integer combination is solved to determine the values of the continuous variables x^k. If the (NLP-I) subproblem is feasible, a valid UB and new OA cuts can be obtained based on its solution; otherwise, the (NLP-f) subproblem is solved to generate only new OA cuts. These new OA cuts are added to all open nodes of the search tree, and the B&B procedure continues with an improved polyhedral outer approximation of the nonlinear constraints. It is worth noting that the LP problem at the node that provided the integer combination y^k should be resolved with the improved polyhedral approximation (i.e., resolved after adding the OA cuts generated based on solution (x^k, y^k)). However, this node is always pruned since its objective value is either equal to or worse than the current best UB. The search does not stop until the difference between the best UB and LB is within the given optimality gap ϵ. The detailed steps of the LP/NLP B&B algorithm are summarized in Algorithm 2.

The LP/NLP B&B method is guaranteed to terminate at the global optimal solution of only convex MINLP problems (Quesada and Grossmann 1992). Using the same tools mentioned in Section 2.2, the LP/NLP B&B method can be extended to handle non-convex problems by replacing the OA cuts with McCormick relaxation-based cuts. The detailed steps of the GLP/NLP B&B algorithm are also summarized in Algorithm 2.

Algorithm 2 LP/NLP B&B and GLP/NLP B&B Algorithms

input: The (MINLP) problem and accepted optimality gap $\epsilon \geq 0$

Initialize: $k = 1$, $LB^0 = -\infty$, $UB^0 = +\infty$;
Solve the continuous relaxation of the (MINLP) problem to obtain an initial solution $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$;
Update $LB^0 = f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$, generate cutting planes at the initial solution $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ according to (cut[1]), and construct the (MILP-master[2]) problem;

while in the B&B tree of the (MILP-master[2]) problem and $UB^{k-1} - LB^{k-1} > \epsilon$ **do**
    if a new incumbent integer solution $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ is found **then**
        if $\hat{y} \in \left\{ y^i \mid i = 1, \ldots, k-1 \right\}$ **then**
            skip the remaining steps and continue the B&B procedure;
        else
            obtain $y^k = \hat{y}$, and take the lower bound of the current B&B tree as LB^k;
        end if
    end if
    Solve the (NLP-I) subproblem with integer variables fixed as $\mathbf{y^k}$;
    if the (NLP-I) subproblem is feasible **then**
        Obtain $\mathbf{x^k}$ and update $UB^k = \min\left(f(\mathbf{x^k}, \mathbf{y^k}), UB^{k-1}\right)$;
    else
        Solve the (NLP-f) feasibility subproblem to obtain $\mathbf{x^k}$ and set $UB^k = UB^{k-1}$;
    end if
    Generate cutting planes at the solution $\mathbf{x^k}, \mathbf{y^k}$ according to (cut[1]) and add them to the current node and all open nodes of the B&B tree of the (MILP-master[2]) problem;
    if GLP/NLP B&B algorithm is selected **then**
        Generate no-good cuts at the solution $\mathbf{x^k}, \mathbf{y^k}$ according to (NoGood-cut) and add them to the current node and all open nodes of the B&B tree of the (GOA-master) problem;
    end if
    $k = k + 1$;
end while

return Best found solution

Note:[1]use (OA-cut) for LP/NLP B&B and (MC-cut) for GLP/NLP B&B. [2]use (OA-master) for LP/NLP B&B and (GOA-master) for GLP/NLP B&B.

3. Convexification- and Bound-Tightening-Based (G)OA and (G)LP/NLP-Based B&B

The success of the multi- and single-tree (G)OA methods, as described in Algorithms 1 and 2, relies on generating tight cuts at the boundary of the nonlinear feasible region defined by the (MINLP) problem. However, these approaches incur the computational cost of solving numerous NLP subproblems. Notably, in the early iterations of the algorithms, only a few linear inequality constraints (i.e., OA and McCormick relaxation-based cuts) are added to the (OA-master) and (GOA-master) problems. As a result, the constructed master problems provide only a poor approximation of the (MINLP) problem; thus, the integer combinations obtained by solving them tend to yield infeasible (NLP-I) subproblems. Although the (NLP-f) feasibility subproblems are solved instead to find points closest to the feasible region and then generate the corresponding tightest cuts, these cuts are generally not as tight as those produced directly at the boundary. Additionally, no valid UB can be obtained in this case. This inefficiency in convergence increases the computational burden. Motivated by this consideration, we propose to integrate (1) bound tightening techniques and (2) pre-generated linear convexification cuts into the (G)OA and (G)LP/NLP B&B methods. The goal is to reduce the occurrence of infeasible (NLP-I) subproblems and accelerate the convergence of the algorithms for both convex and non-convex MINLP problems. The details of the proposed methods are explained below.

Bound tightening is a crucial component of domain reduction techniques. There are two main approaches: feasibility-based bound tightening (FBBT) and optimality-based bound tightening (OBBT). They have been proven effective in accelerating the speed of convergence in many optimization solvers by reducing variable domains and eliminating infeasible and inferior subdomains of the search space (Belotti et al. 2009, Ryoo and Sahinidis 1996). The FBBT techniques primarily rely on interval arithmetic and systematic exploitation of relationships between problem constraints and variable bounds (i.e., through forward and back propagation) (Belotti et al. 2012). As a result, infeasible subdomains of the search space can be eliminated while all feasible solutions are preserved. In contrast, the OBBT techniques typically require solving auxiliary optimization problems for each variable to compute the corresponding tightest bounds. These auxiliary problems are generally the convex relaxations of the original problem, where the objective is to minimize or maximize the single variable subject to all problem constraints (Gleixner et al. 2017). When an objective cut-off (e.g., using the best-known UB) is incorporated, the OBBT techniques can further eliminate the feasible points from the search space while ensuring that at least one globally optimal solution remains. One related technique, often referred to in the literature as marginals-based bound tightening (MBBT), is mainly used in specific contexts where dual variable information is available (i.e., obtained from solving the convex relaxation of the original problem). It can also cut off feasible points that have worse objective values than the current best UB. In B&B-based methods, these bound-tightening techniques not only reduce the size of the search domain but also strengthen the convex relaxation of the original problem. This usually leads to better LBs and thus drastically facilitates node pruning. We think that the (G)OA and (G)LP/NLP B&B methods could benefit from the advantages of reduced variable domains achieved by FBBT, OBBT, and MBBT. Specifically, tighter bounds can improve the polyhedral outer approximation of the original feasible region for MILP master problems, particularly during early iterations. For NLP subproblems, tighter bounds can decrease computational time by narrowing the search space to be explored. However, it is important to acknowledge that implementing these bound-tightening techniques can be computationally expensive. In certain circumstances, solving auxiliary problems to compute tighter variable bounds might offset any gains in the spend of convergence. To manage this trade-off, we implement these techniques selectively at the root node of the (G)OA and (G)LP/NLP B&B algorithms (i.e., their initialization stage), which generate the following tighter constraints for the integer and continuous variables.

$$x \in \left[\underline{x}', \overline{x}'\right] \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n \tag{1}$$

$$y \in \left\{\underline{y}', \ldots, \overline{y}'\right\} \in \left\{\underline{y}, \ldots, \overline{y}\right\} \subseteq \mathbb{Z}^m \tag{2}$$

Convexification techniques are widely employed to address the computational challenges posed by non-convexities. These methods are primarily used to construct valid and converging convex and concave relaxations of the non-convex constraints in the original problem, thereby reformulating it into a structure suitable for convex optimization solvers and providing a valid LB. Among these techniques, AVM is a general method to handle non-convex factorable functions. Specifically, it introduces an auxiliary variable and a corresponding auxiliary equality constraint for each intermediate nonlinear factor of the given function. These introduced auxiliary variables are generally defined by simple and low dimensional functions, making it computationally efficient to construct their convex and concave relaxations. Then, the affine under and overestimators for these convex and concave relaxations are computed at selected linearization points using their subgradient information. With these, a set of linear convexification cuts can be derived and generalized as follows, which further refines the polyhedral outer approximation of the feasible region defined by the original problem.

$$Dx + Ey + Fz \leq d \tag{3}$$

Herein, the AVM is also implemented at the initialization stage of the (G)OA and (G)LP/NLP B&B algorithms to generate the linear convexification cuts. While these cuts are theoretically redundant for NLP subproblems where the original nonlinear constraints are explicitly considered, they can help provide a significantly improved polyhedral outer approximation of the original feasible region at the beginning of the algorithms. It is worth noting that, for each introduced auxiliary variable, additional constraints for the convex and concave relaxation of the auxiliary equality constraint must be incorporated into the original problem. Thus, the total number of the additional linear inequality constraints is ultimately determined by the number of the selected linearization points. The potential efficiency gains come at the cost of introducing many auxiliary variables and their corresponding linear inequality constraints, which may increase the overall computational requirement.

Using the tightened variable bounds and the linear convexification cuts, the (OA-master) problem is reformulated as the (C-BT-OA-master) problem as follows. Similarly, the (GOA-master) problem is reformulated as the (C-BT-GOA-master) problem.

$$
\begin{aligned}
\min_{x,y,\mu} \quad & \mu \\
s.t. \quad & f\left(x^i, y^i\right) + \nabla f\left(x^i, y^i\right)^{\mathrm{T}} \begin{bmatrix} x - x^i \\ y - y^i \end{bmatrix} \le \mu \quad \forall i = 1, \dots, k \\
& g_j\left(x^i, y^i\right) + \nabla g_j\left(x^i, y^i\right)^{\mathrm{T}} \begin{bmatrix} x - x^i \\ y - y^i \end{bmatrix} \le 0 \quad \forall i = 1, \dots, k, \forall j \in I_i \\
& Ax + By \le b, \\
& Dx + Ey + Fz \le d \\
& x \in \left[\underline{x}', \overline{x}'\right] \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n \\
& y \in \left\{\underline{y}', \dots, \overline{y}'\right\} \in \left\{\underline{y}, \dots, \overline{y}\right\} \subseteq \mathbb{Z}^m \\
& \mu \in \mathbb{R}
\end{aligned}
\tag{C-BT-OA-master}
$$

$$
\begin{aligned}
\min_{x,y,\mu} \quad & \mu \\
s.t. \quad & f^{cv}\left(x^i, y^i\right) + \left(s_f^{cv}\left(x^i, y^i\right)\right)^{\mathrm{T}} \begin{bmatrix} x - x^i \\ y - y^i \end{bmatrix} \le \mu \quad \forall i = 1, \dots, k \\
& g_j^{cv}\left(x^i, y^i\right) + \left(s_{g_j}^{cv}\left(x^i, y^i\right)\right)^{\mathrm{T}} \begin{bmatrix} x - x^i \\ y - y^i \end{bmatrix} \le 0 \quad \forall i = 1, \dots, k, \forall j \in I_i \\
& Ax + By \le b, \\
& \sum_{j \in V_1^i} y_j - \sum_{j \in V_0^i} y_j \le |V_1^i| - 1 \quad \forall i = 1, \dots, k \\
& Dx + Ey + Fz \le d \\
& x \in \left[\underline{x}', \overline{x}'\right] \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n \\
& y \in \left\{\underline{y}', \dots, \overline{y}'\right\} \in \left\{\underline{y}, \dots, \overline{y}\right\} \subseteq \mathbb{Z}^m \\
& \mu \in \mathbb{R}
\end{aligned}
\tag{C-BT-GOA-master}
$$

The (NLP-I) and (NLP-f) subproblems are reformulated as the corresponding (C-BT-NLP-I) and (C-BT-NLP-f) subproblems.

$$\begin{aligned} \min_{x} \quad & f(x, y^{k+1}) \\ s.t. \quad & g_j\left(x, y^{k+1}\right) \leq 0 \quad \forall j = 1, \ldots, l \\ & Ax + By^{k+1} \leq b, \\ & Dx + Ey + Fz \leq d \\ & x \in \left[\underline{x}', \overline{x}'\right] \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n \end{aligned} \tag{C-BT-NLP-I}$$

$$\begin{aligned} \min_{x,s} \quad & \|s\|_p \\ s.t. \quad & g_j\left(x, y^{k+1}\right) \leq s_j \quad \forall j = 1, \ldots, l \\ & Ax + By^{k+1} \leq b, \\ & Dx + Ey + Fz \leq d \\ & x \in \left[\underline{x}', \overline{x}'\right] \in \left[\underline{x}, \overline{x}\right] \subseteq \mathbb{R}^n \end{aligned} \tag{C-BT-NLP-f}$$

Considering that the convexification cuts are redundant in the (C-BT-NLP-I) and (C-BT-NLP-f) subproblems, we denote the (C-BT-NLP-I) and (C-BT-NLP-f) subproblems as complete-scale NLP problems; while the subproblems without the convexification cuts are denoted as reduced-scale NLP problems.

Similar modifications are also applied to the GOA, LP/NLP-B&B, and GLP/NLP-B&B methods for both convex and non-convex MINLP problems. The detailed steps of these convexification- and bound-tightening-based algorithms are summarized in Algorithms 3 and 4, which are referred to as C-BT-(G)OA and C-BT-(G)LP/NLP B&B Algorithms. The major contributions are highlighted in red.

4. Benchmarking and Results

To evaluate the impact of convexification and domain reduction techniques, we use test instances from the MINLPLib problem library (Vigerske 2014). Specifically, 434 convex instances and 181 non-convex instances are selected, adhering to the criteria that each instance must have at least one discrete variable and at least one continuous variable. For clarity in our analysis, we use (r) and (c) to

Algorithm 3 C-BT-OA and C-BT-GOA Algorithms

input: The (MINLP) problem and accepted optimality gap $\epsilon \geq 0$

Initialize: $k = 1$, $LB^0 = -\infty$, $UB^0 = +\infty$;

Initialize (C-BT-): Apply bound tightening techniques to obtain the tightened variable bounds, apply convexification techniques to obtain the linear convexification cuts, and add them to the (MINLP) problem;

Solve the continuous relaxation of the (MINLP) problem to obtain an initial solution $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$;

Update $LB^0 = f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$, generate cutting planes at the initial solution $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ according to (cut[1]), and construct the (MILP-master[2]) problem;

while $UB^{k-1} - LB^{k-1} > \epsilon$ **do**

  Solve the (MILP-master[2]) problem to obtain $\mathbf{y^k}$ and LB^k;

  Solve the (C-BT-NLP-I) subproblem with integer variables fixed as $\mathbf{y^k}$;

  if the (C-BT-NLP-I) subproblem is feasible **then**

    Obtain $\mathbf{x^k}$ and update $UB^k = \min\left(f(\mathbf{x^k}, \mathbf{y^k}), UB^{k-1}\right)$;

  else

    Solve the (C-BT-NLP-f) feasibility subproblem to obtain $\mathbf{x^k}$ and set $UB^k = UB^{k-1}$;

  end if

  Generate cutting planes at the solution $\mathbf{x^k}, \mathbf{y^k}$ according to (cut[1]) and add them to the (MILP-master[2]) problem;

  if C-BT-GOA algorithm algorithm is selected **then**

    Generate no-good cuts at the solution $\mathbf{x^k}, \mathbf{y^k}$ according to (NoGood-cut) and add them to the (C-BT-GOA-master) problem;

  end if

  $k = k + 1$;

end while

if C-BT-GOA algorithm is selected **then**

  Solve the (C-BT-GOA-master) problem without the no-good cuts to obtain the valid LB of the (MINLP) problem;

end if

return Best found solution

Note:[1]use (OA-cut) for C-BT-OAand (MC-cut) for C-BT-GOA. [2]use (OA-master) for C-BT-OA and (GOA-master) for C-BT-GOA.

Algorithm 4 LP/NLP B&B and GLP/NLP B&B Algorithms

input: The (MINLP) problem and accepted optimality gap $\epsilon \geq 0$

Initialize: $k = 1$, $LB^0 = -\infty$, $UB^0 = +\infty$;

Initialize (C-BT-): Apply bound tightening techniques to obtain the tightened variable bounds, apply convexification techniques to obtain the linear convexification cuts, and add them to the (MINLP) problem;

Solve the continuous relaxation of the (MINLP) problem to obtain an initial solution $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$;

Update $LB^0 = f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$, generate cutting planes at the initial solution $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ according to (cut[1]), and construct the (MILP-master[2]) problem;

while in the B&B tree of the (MILP-master[2]) problem and $UB^{k-1} - LB^{k-1} > \epsilon$ **do**

    if a new incumbent integer solution $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ is found **then**

        if $\hat{y} \in \left\{ y^i \mid i = 1, \ldots, k-1 \right\}$ **then**

            skip the remaining steps and continue the B&B procedure;

        else

            obtain $y^k = \hat{y}$, and take the lower bound of the current B&B tree as LB^k;

        end if

    end if

    Solve the (C-BT-NLP-I) subproblem with integer variables fixed as $\mathbf{y^k}$;

    if the (C-BT-NLP-I) subproblem is feasible **then**

        Obtain $\mathbf{x^k}$ and update $UB^k = \min\left(f(\mathbf{x^k}, \mathbf{y^k}), UB^{k-1}\right)$;

    else

        Solve the (C-BT-NLP-f) feasibility subproblem to obtain $\mathbf{x^k}$ and set $UB^k = UB^{k-1}$;

    end if

    Generate cutting planes at the solution $\mathbf{x^k}, \mathbf{y^k}$ according to (cut[1]) and add them to the current node and all open nodes of the B&B tree of the (MILP-master[2]) problem;

    if C-BT-GLP/NLP B&B algorithm is selected **then**

        Generate no-good cuts at the solution $\mathbf{x^k}, \mathbf{y^k}$ according to (NoGood-cut) and add them to the current node and all open nodes of the B&B tree of the (C-BT-GOA-master) problem;

    end if

    $k = k + 1$;

end while

return Best found solution

Note:[1]use (OA-cut) for C-BT-LP/NLP B&B and (MC-cut) for C-BT-GLP/NLP B&B. [2]use (OA-master) for C-BT-LP/NLP B&B and (GOA-master) for C-BT-GLP/NLP B&B.

distinguish between the reduced- and complete-scale NLP subproblems used in the convexification- and bound-tightening-based OA and LP/NLP-B&B methods. Moreover, we indicate the results with convexification cuts considered with the prefix (C-). The benchmark implementation is based on the Mixed-integer nonlinear decomposition toolbox for Pyomo - MindtPy (Bernal et al. 2018). We evaluate both the multi-tree and single-tree settings of the OA and GOA strategies, maintaining their default configurations as the baseline. Furthermore, we integrate two implementations of the convexification and BT techniques for MINLP problems in MindtPy. The first implementation is based on a particular version of BARON 19.4.4, the state-of-the-art commercial MINLP solver. Instead of completely solving the MINLP model, we use the particular version of BARON to preprocess it and obtain an MINLP with tightened bounds and pre-generated convexification cuts. Both options `dolocal` and `numloc` are set to 0 to turn off local search during upper bounding and preprocessing in BARON. All range reduction and relaxation options retain their default settings. Thus, linear FBBT, nonlinear FBBT, and MBBT are applied, as well as the outer approximations of convex univariate functions and cutting planes. The second implementation uses Coramin (Bynum et al. 2020) and the FBBT (C++) code in Pyomo, both of which are open-source and offer more flexibility. We evaluate the enhancements brought about by the convexification cuts and FBBT to the OA and LP/NLP B&B methods, both individually and in combination. When combined, the nonlinear FBBT is first applied once, followed by Coramin to perform the AVM and generate the initial OA cuts based on the reformulation at the initialization stage. This implementation is accessible via the `mindtpy-coa-coramin` branch of Pyomo. It offers three configurable options: `use_fbbt`, `use_obbt`, and `use_convex_relaxation`, which enable FBBT, OBBT, and convexification cuts, respectively. For the algorithm termination criteria, we set the absolute optimality tolerances $\epsilon = 10^{-5}$ and the relative optimality tolerance $\epsilon_{rel} = 10^{-3}$, along with a time limit of 900 seconds. We use GUROBI 11.0.2 as the MILP solver, IPOPTH 3.14 as the NLP solver for convex instances, and BARON 23.6.22 as the NLP solver for non-convex instances. All tests ran on a Linux cluster with 48 AMD EPYC 7643 2.3GHz CPUs and 1 TB RAM, with each test restricted to using only a single thread. The PAVER report of the benchmark results is available at https://secquoia.github.io/Convexification-based-OA-Benchmark.

The comparison of the number of instances solved to optimality between the proposed methods and baseline methods is presented in Table 1. The percentage in the table represents the improvement of the proposed method over the corresponding baseline method. For the C-LP/NLP-B&B-BARON(r) and C-LP/NLP-B&B-BARON(c) method, it is worth noting that the convexification and bound tightening processes in BARON result in IPOPTH failure in the `st_test1`, `st_test2`, `st_testgr1`, `st_testgr3`, `st_testgr4` instances (Tawarmalani and Sahinidis 2013, Shectman 1999). This makes both these methods underperform the baseline. Overall, the convexification and bound tightening techniques are beneficial to all the OA (+6.6%), GOA (+21.8%), LP/NLP-B&B (+0.9%), and GLP/NLP-B&B (+36.5%) methods, helping to solve more instances to optimality within the timelimit.

The time and iteration performance profiles of the (C-)OA methods and the (C-)LP/NLP-B&B methods for the convex instances are presented in Figure 1 - 4. Particularly, regarding the LP/NLP-B&B method, the number of iterations refers to the number of subproblems (NLP-I) solved. Overall, the C-OA methods, regardless of the convexification and FBBT implementations used, outperform the standard OA method. The C-OA-BARON(r) method emerges as the best, demonstrating that the presolve implementation in BARON is more efficient than those in Coramin and the FBBT code in Pyomo, as expected. The C-OA-BARON(r) method and the C-OA-BARON(c) method achieve similar performance in terms of iterations, while the C-OA-BARON(r) method is superior in solution time. This observation supports our previous assertion that the convexification cuts are redundant and increase the computational complexity of subproblems (NLP-I) and (NLP-f). This is further illustrated by the benchmark results of the (C-)LP/NLP-B&B methods in Figures 3 and 4, where the C-LP/NLP-B&B-BARON(c) method underperforms in solution time compared with the standard LP/NLP-B&B method. The deterioration results from the fact that generally, more NLP subproblems are solved in the LP/NLP-B&B method compared with the OA method, combined with the IPOPTH failure induced by BARON presolve. Since the choice between complete-scale and reduced-scale NLP subproblems only increases the complexity of the NLP subproblems, there is no significant difference in the iteration performance between the C-OA-BARON(r) and C-OA-BARON(c) methods.

When FBBT is applied alone, neither the OA method nor the LP/NLP-B&B method benefits in terms of the solution time and iterations. However, FBBT can improve performance when combined with convexification cuts. In the iteration profiles, both the C-OA-BARON(r) and C-OA-BARON(c) methods outperform the C-OA-FBBT-Coramin(r) method, which means that BARON presolve generates tighter relaxations than Coramin. For simple instances solvable within 5 seconds, both the standard OA method and the LP/NLP B&B method are more efficient than their convexification-based counterparts due to the additional processing time required for both the bound tightening and the generation of convexification cuts. However, bound tightening and convexification techniques

Table 1 Comparison of instances solved to optimality: proposed methods vs. baseline methods

Method	Number of instances solved to optimality
OA	319
OA-FBBT	320 (+0.3%)
C-OA-Coramin(r)	334 (+4.7%)
C-OA-FBBT-Coramin(r)	338 (+6.0%)
C-OA-BARON(r)	340 (+6.6%)
C-OA-BARON(c)	338 (+6.0%)
LP/NLP-B&B	347
LP/NLP-B&B-FBBT	345 (-0.6%)
C-LP/NLP-B&B-Coramin(r)	350 (+0.9%)
C-LP/NLP-B&B-FBBT-Coramin(r)	348 (+0.3%)
C-LP/NLP-B&B-BARON(r)	346 (-0.3%)
C-LP/NLP-B&B-BARON(c)	335 (-3.5%)
GOA	55
GOA-FBBT	61 (+10.9%)
C-GOA-Coramin(r)	64 (+16.4%)
C-GOA-FBBT-Coramin(r)	67 (+21.8%)
C-GOA-BARON(r)	67 (+21.8%)
C-GOA-BARON(c)	65 (+18.2%)
GLP/NLP-B&B	52
GLP/NLP-B&B-FBBT	57 (+9.6%)
C-GLP/NLP-B&B-Coramin(r)	68 (+30.8%)
C-GLP/NLP-B&B-FBBT-Coramin(r)	70 (+34.6%)
C-GLP/NLP-B&B-BARON(r)	71 (+36.5%)
C-GLP/NLP-B&B-BARON(c)	68 (+30.8%)

generally enhance the performance of both the OA and LP/NLP B&B methods for convex MINLP instances.

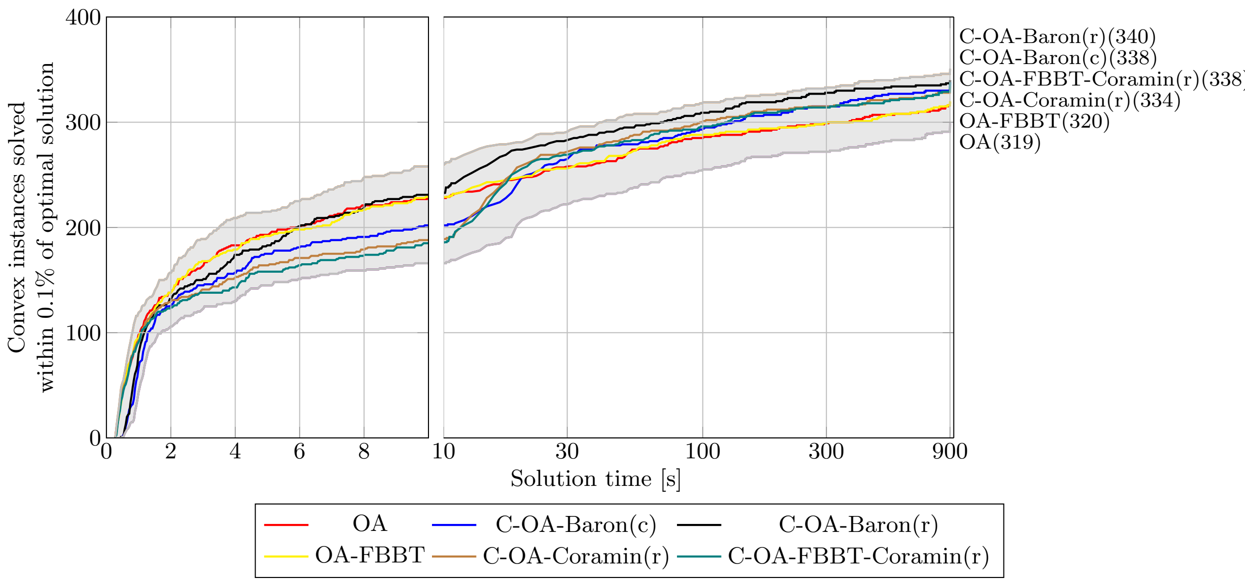

Figure 1 **Time performance profile of the (C-)OA method for convex instances**

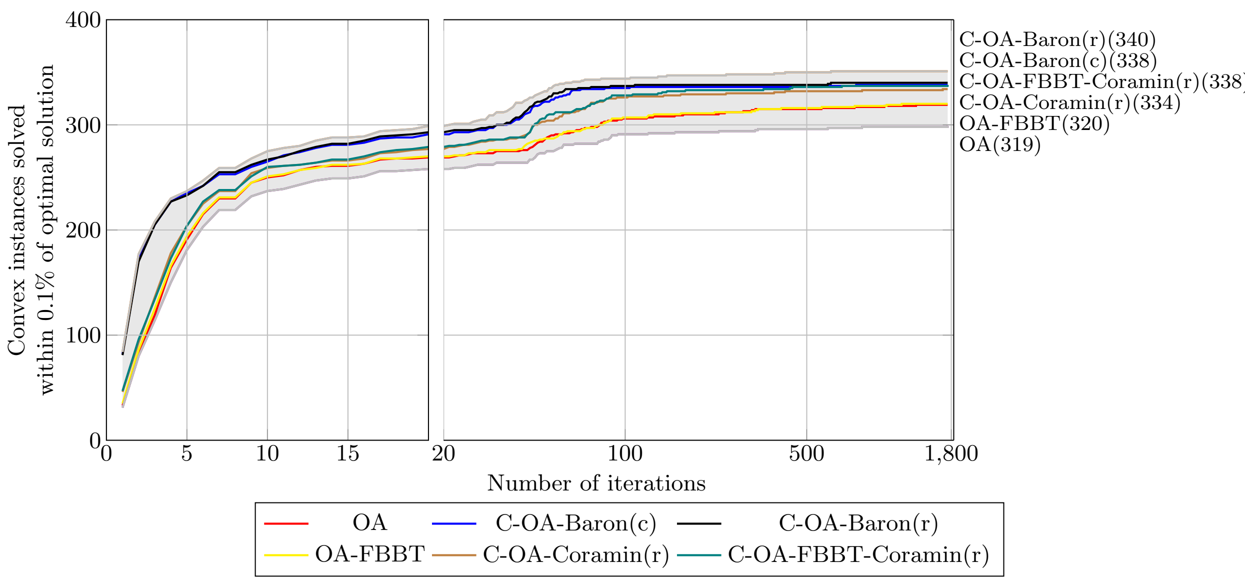

Figure 2 **Iteration performance profile of the (C-)OA method for convex instances**

The time and iteration performance profiles of the GOA method and the GLP/NLP-B&B method for non-convex instances are presented in Figures 5 - 8. For non-convex problems, it is common that the algorithm finds the optimal solution but cannot close the gap within the time limit, resulting in noticeable jumps at 900 seconds in Figures 5 and 7. Like the convex cases, the convexification-based GOA and GLP/NLP B&B methods utilizing reduced-scale NLP subproblems outperform their standard counterparts in solution time and number of iterations. The advantage is even more pronounced compared to the convex benchmark results. However, in contrast to the convex benchmark

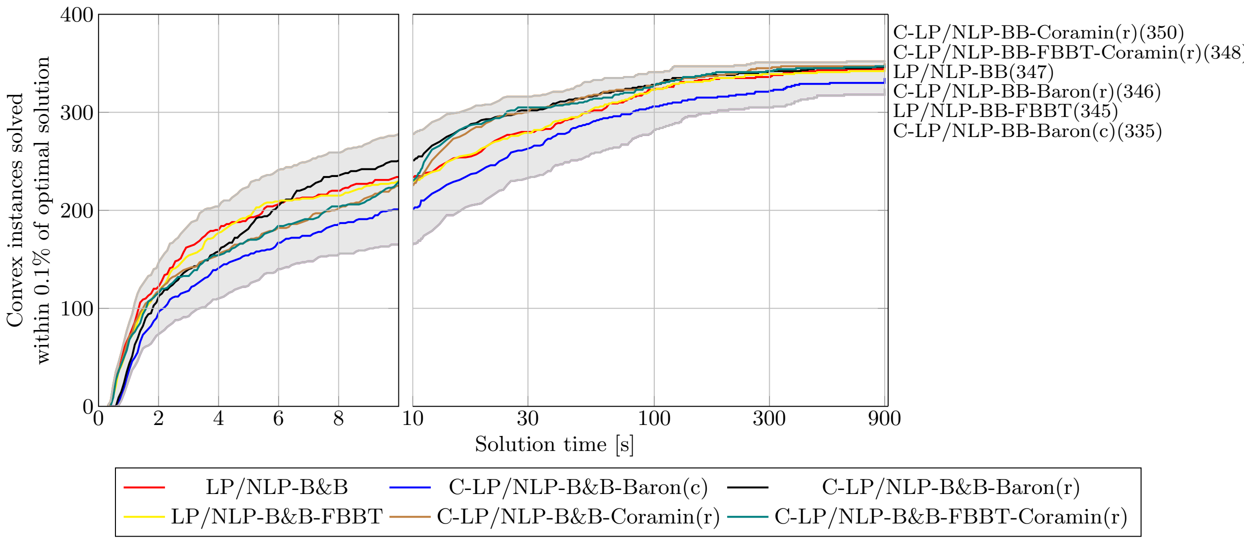

Figure 3 Time performance profile of the (C-)LP/NLP-B&B method for convex instances

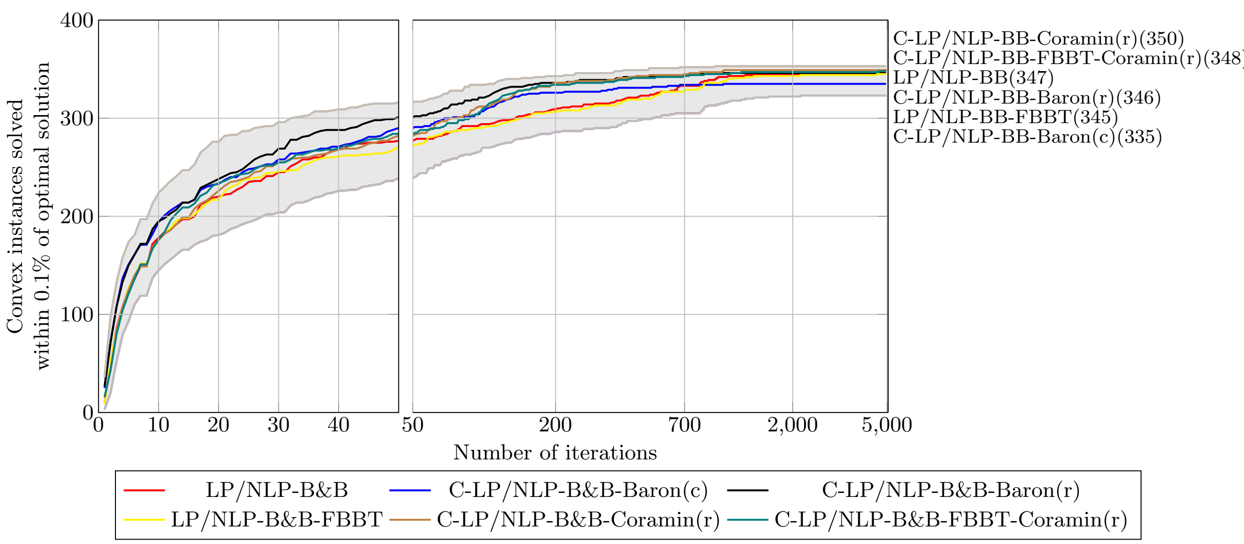

Figure 4 Iteration performance profile of the (C-)LP/NLP-B&B method for convex instances

results, we noted that FBBT is more beneficial to both the GOA method and the GLP/NLP method for non-convex instances, with significant improvements observed when FBBT is applied alone. As we know, non-convex MINLP instances are generally more challenging to solve. We noticed that the standard GOA method and the GLP/NLP B&B method are less efficient even for simple MINLP instances, highlighting the importance of applying bound tightening and convexification techniques even for small non-convex MINLP instances. Overall, consistent performance in convex and non-convex MINLP problems demonstrates the effectiveness of the proposed convexification-based OA and LP/NLP B&B methods.

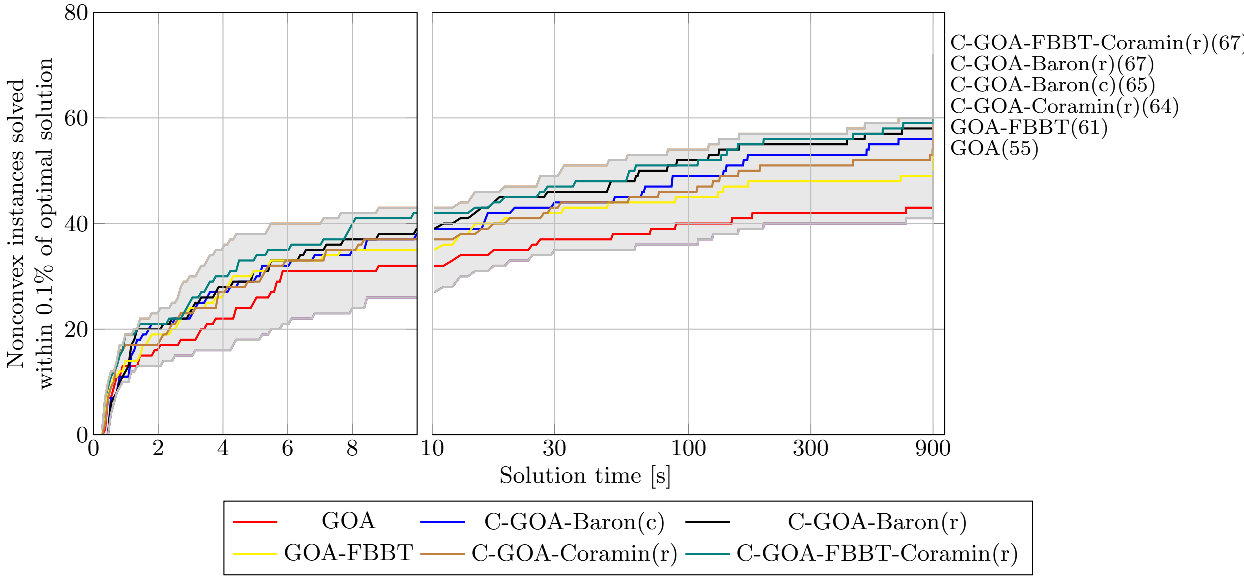

Figure 5 **Time performance profile of the (C-)GOA method for convex instances**

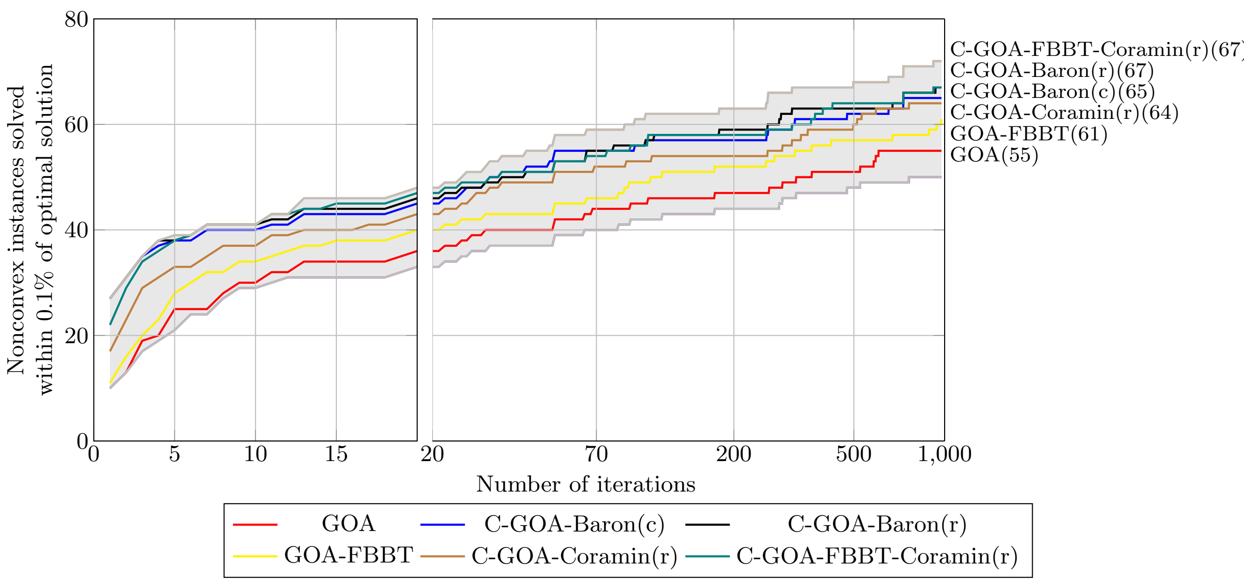

Figure 6 **Iteration performance profile of the (C-)GOA method for convex instances**

The performance of the proposed methods on large instances is further analyzed in Table 2, considering solution time, number of iterations, and number of infeasible (NLP-I) subproblems. Instances that failed or were solved within 10 seconds by the baseline methods are excluded from the analysis. The percentages in the table represent the improvement of the proposed method over the corresponding baseline method. The performance index of each method is calculated using the shifted geometric mean, with a shift value of 10, consistent with the Mittelmann benchmarks. Among the four group methods, the BARON-based convexification and bound-tightening imple-

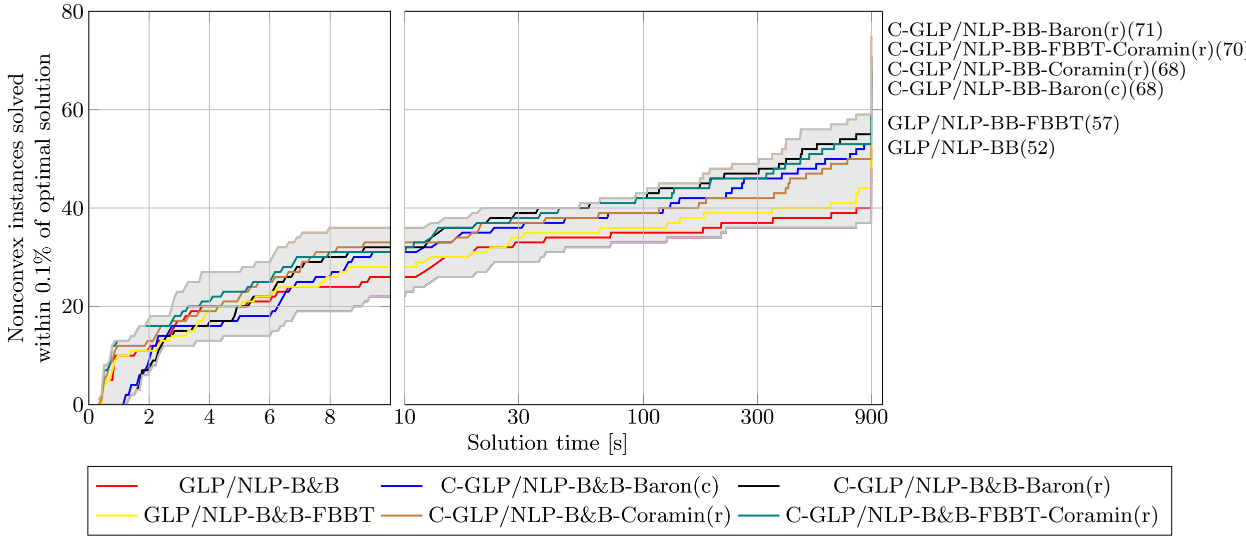

Figure 7 **Time performance profile of the (C-)GLP/NLP-B&B method for non-convex instances**

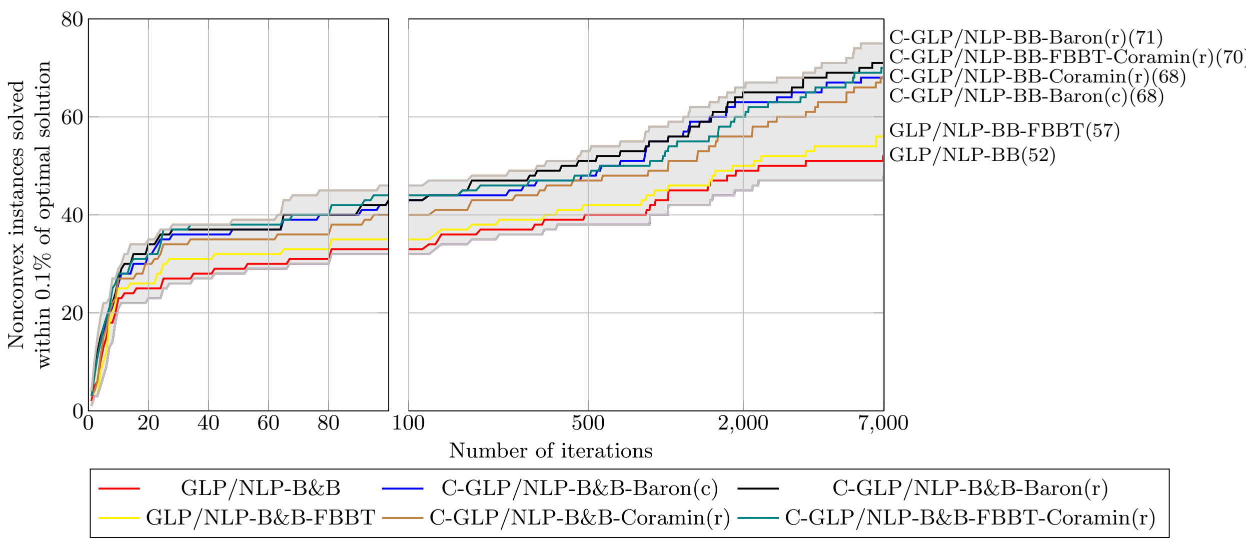

Figure 8 **Iteration performance profile of the (C-)GLP/NLP-B&B method for non-convex instances**

mentation with reduced-scale (NLP-I) subproblem performs the best on large instances overall. Additionally, we observe that the number of infeasible (NLP-I) subproblems can be significantly reduced by more than 60% when convexification and bound-tightening techniques are applied. This reduction of infeasible subproblems allows stronger cuts to be generated at the early stage and less time to be spent solving the feasibility subproblems. Consequently, these factors contribute to the algorithm converging in fewer iterations and less time.

Table 2 Performance indexes of the proposed methods on large instances

Method	Solution time	# of iterations	# of infeasible subproblems
OA	126.8	29	7.6
OA-FBBT	130.1 (+2.65%)	28.8 (-0.97%)	7.5 (-0.46%)
C-OA-Coramin(r)	121.4 (-4.26%)	21.3 (-26.53%)	5.4 (-28.12%)
C-OA-FBBT-Coramin(r)	130.7 (+3.15%)	21.8 (-24.94%)	5.4 (-28.74%)
C-OA-BARON(r)	93.6 (-26.18%)	13.4 (-53.74%)	2.0 (-73.71%)
C-OA-BARON(c)	102.2 (-19.37%)	13.2 (-54.52%)	1.9 (-74.73%)
LP/NLP-B&B	71.2	121.9	9.8
LP/NLP-B&B-FBBT	64.8 (-9.01%)	120.1 (-1.47%)	10.0 (+1.49%)
C-LP/NLP-B&B-Coramin(r)	51.1 (-28.26%)	78.5 (-35.66%)	7.5 (-23.84%)
C-LP/NLP-B&B-FBBT-Coramin(r)	50.2 (-29.57%)	75.0 (-38.52%)	7.6 (-22.73%)
C-LP/NLP-B&B-BARON(r)	43.4 (-39.05%)	49.7 (-59.25%)	3.6 (-62.82%)
C-LP/NLP-B&B-BARON(c)	57.5 (-19.33%)	50.0 (-58.98%)	3.5 (-64.64%)
GOA	587.2	59	12
GOA-FBBT	518.1 (-11.76%)	53.6 (-9.20%)	8.7 (-27.90%)
C-GOA-Coramin(r)	511.2 (-12.94%)	41.6 (-29.47%)	7.7 (-35.77%)
C-GOA-FBBT-Coramin(r)	454.9 (-22.53%)	41.0 (-30.44%)	6.2 (-48.44%)
C-GOA-BARON(r)	448.3 (-23.65%)	40.4 (-31.48%)	6.1 (-49.45%)
C-GOA-BARON(c)	444.4 (-24.31%)	39.9 (-32.33%)	6.4 (-46.38%)
GLP/NLP-B&B	671.3	227.5	17.2
GLP/NLP-B&B-FBBT	660.3 (-1.64%)	236.8 (+4.10%)	16.6 (-3.52%)
C-GLP/NLP-B&B-Coramin(r)	616.2 (-8.21%)	163.1 (-28.29%)	9.6 (-44.44%)
C-GLP/NLP-B&B-FBBT-Coramin(r)	592.5 (-11.74%)	164.0 (-27.89%)	7.5 (-56.44%)
C-GLP/NLP-B&B-BARON(r)	554.4 (-17.42%)	151.5 (-33.42%)	6.9 (-59.58%)
C-GLP/NLP-B&B-BARON(c)	575.7 (-14.24%)	128.7 (-43.43%)	6.2 (-63.72%)

5. Conclusion

This work explores the impact of convexification and bound tightening techniques implemented in B&B-based solvers on the OA and the LP/NLP-B&B methods. These effects were investigated for variations of these methods in solving both convex and non-convex MINLPs to global optimality. The proposed convexification- and bound-tightening-based OA and LP/NLP-B&B methods are implemented within the open-source solver MindtPy. Our benchmarking results highlight significant improvements by domain reduction techniques in enhancing the efficiency of the OA and LP/NLP-

B&B methods, observed by reducing computational times and the number of iterations required to solve convex and non-convex MINLP problems to global optimality. These results highlight the value of implementing domain reduction techniques, which are successful for B&B methods in MINLP decomposition algorithms. Additionally, the idea of tightening the feasible region of the master problem at the presolve stage can also be applied to other cutting plane methods and cutting-plane-based decomposition methods, such as Benders decomposition and generalized Benders decomposition methods. An interesting direction is how to generate effective and reasonable-scale cuts to accelerate the convergence of these iterative methods.

Acknowledgments

D.B.N. and Z.P. acknowledge the support of the startup grant of the Davidson School of Chemical Engineering at Purdue University. The authors acknowledge the support of Prof. Nick Sahinidis in providing a special version of BARON and Dr. Bynum Michael Lee in providing Coramin, which allowed for this research. The authors thank GAMS Development Corp. for providing a license to their software and various optimization solvers used for this manuscript.

References

Achterberg T (2009) SCIP: solving constraint integer programs. *Mathematical Programming Computation* 1:1–41.

Adjiman CS, Floudas CA (1996) Rigorous convex underestimators for general twice-differentiable problems. *Journal of Global Optimization* 9:23–40.

Akrotirianakis IG, Floudas CA (2004) A new class of improved convex underestimators for twice continuously differentiable constrained nlps. *Journal of Global Optimization* 30:367–390.

Bacci T, Frangioni A, Gentile C, Tavlaridis-Gyparakis K (2024) New mixed-integer nonlinear programming formulations for the unit commitment problems with ramping constraints. *Operations Research* 72(5):2153–2167.

Belotti P, Cafieri S, Lee J, Liberti L (2012) On feasibility based bounds tightening .

Belotti P, Lee J, Liberti L, Margot F, Wächter A (2009) Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software* 24(4-5):597–634.

Bernal DE, Chen Q, Gong F, Grossmann IE (2018) Mixed-integer nonlinear decomposition toolbox for pyomo (MindtPy). *Computer Aided Chemical Engineering*, volume 44, 895–900 (Elsevier).

Bernal DE, Peng Z, Kronqvist J, Grossmann IE (2022) Alternative regularizations for outer-approximation algorithms for convex MINLP. *Journal of Global Optimization* 84(4):807–842.

Bernal DE, Vigerske S, Trespalacios F, Grossmann IE (2020) Improving the performance of DICOPT in convex MINLP problems using a feasibility pump. *Optimization Methods and Software* 35(1):171–190.

Bestuzheva K, Chmiela A, Müller B, Serrano F, Vigerske S, Wegscheider F (2023) Global optimization of mixed-integer nonlinear programs with SCIP 8. *Journal of Global Optimization* 1–24.

Bonami P, Biegler LT, Conn AR, Cornuéjols G, Grossmann IE, Laird CD, Lee J, Lodi A, Margot F, Sawaya N, et al. (2008) An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* 5(2):186–204.

Bonami P, Cornuéjols G, Lodi A, Margot F (2009) A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming* 119(2):331–352.

Bonami P, Lejeune MA (2009) An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Operations research* 57(3):650–670.

Bragalli C, D'Ambrosio C, Lee J, Lodi A, Toth P (2012) On the optimal design of water distribution networks: a practical MINLP approach. *Optimization and Engineering* 13:219–246.

Bynum ML, Castillo A, Ceccon F, Knueven B, Laird CD, Siirola JD (2020) Coramin: An open-source toolset for developing MINLP algorithms. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Chachuat B (2013) MC++: Toolkit for construction, manipulation and bounding of factorable functions. URL https://github.com/omega-icl/mcpp.

Coey C, Lubin M, Vielma JP (2020) Outer approximation with conic certificates for mixed-integer convex problems. *Mathematical Programming Computation* 12(2):249–293.

Dakin RJ (1965) A tree-search algorithm for mixed integer programming problems. *The Computer Journal* 8(3):250–255.

Duran MA, Grossmann IE (1986) An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming* 36:307–339.

Floudas CA (2013) *Deterministic global optimization: theory, methods and applications*, volume 37 (Springer Science & Business Media).

Furman K (2006) Global optimization of deterministic and stochastic nonconvex MINLP via outer approximation. Invited Talk.

Furman K, Wei J, Duran M, Tawarmalani M, Sahinidis N (2005) Global optimization of nonconvex stochastic mixed-integer nonlinear programming (MINLP) problems. *INFORMS Annual Meeting 2005* (San Francisco, CA).

Geoffrion AM (1972) Generalized benders decomposition. *Journal of Optimization Theory and Applications* 10:237–260.

Gleixner AM, Berthold T, Müller B, Weltge S (2017) Three enhancements for optimization-based bound tightening. *Journal of Global Optimization* 67:731–757.

Kesavan P, Allgor RJ, Gatzke EP, Barton PI (2004) Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs. *Mathematical Programming* 100:517–535.

Kocis GR, Grossmann IE (1988) Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis. *Industrial & Engineering Chemistry Research* 27(8):1407–1421.

Kocis GR, Grossmann IE (1989) Computational experience with DICOPT solving MINLP problems in process systems engineering. *Computers & Chemical Engineering* 13(3):307–315.

Kronqvist J, Lundell A, Westerlund T (2016) The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization* 64:249–272.

Lundell A, Kronqvist J, Westerlund T (2022) The supporting hyperplane optimization toolkit for convex MINLP. *Journal of Global Optimization* 84(1):1–41.

Luo ZQ, Ma WK, So AMC, Ye Y, Zhang S (2010) Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine* 27(3):20–34.

Maranas CD, Floudas CA (1992) A global optimization approach for Lennard-Jones microclusters. *The Journal of Chemical Physics* 97(10):7667–7678.

McCormick GP (1976) Computability of global solutions to factorable nonconvex programs: Part I—convex underestimating problems. *Mathematical Programming* 10(1):147–175.

Misener R, Floudas CA (2014) ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations. *Journal of Global Optimization* 59(2):503–526.

Mitsos A, Chachuat B, Barton PI (2009) McCormick-based relaxations of algorithms. *SIAM Journal on Optimization* 20(2):573–601.

Muts P, Nowak I, Hendrix EM (2020) The decomposition-based outer approximation algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization* 77:75–96.

Puranik Y, Sahinidis NV (2017) Domain reduction techniques for global NLP and MINLP optimization. *Constraints* 22(3):338–376.

Quesada I, Grossmann IE (1992) An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers & Chemical Engineering* 16(10-11):937–947.

Ryoo HS, Sahinidis NV (1996) A branch-and-reduce approach to global optimization. *Journal of Global Optimization* 8:107–138.

Sahinidis N, Grossmann IE (1991) MINLP model for cyclic multiproduct scheduling on continuous parallel lines. *Computers & Chemical Engineering* 15(2):85–103.

Sahinidis NV (1996) BARON: A general purpose global optimization software package. *Journal of Global Optimization* 8:201–205.

Shectman JP (1999) *Finite algorithms for global optimization of concave programs and general quadratic programs* (University of Illinois at Urbana-Champaign).

Slater M (2013) Lagrange multipliers revisited. *Traces and emergence of nonlinear programming*, 293–306 (Springer).

Smith EM, Pantelides CC (1997) Global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering* 21:S791–S796.

Smith EM, Pantelides CC (1999) A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex minlps. *Computers & Chemical Engineering* 23(4-5):457–478.

Su L, Tang L, Bernal DE, Grossmann IE (2018) Improved quadratic cuts for convex mixed-integer nonlinear programs. *Computers & Chemical Engineering* 109:77–95.

Tawarmalani M, Sahinidis NV (2001) Semidefinite relaxations of fractional programs via novel convexification techniques. *Journal of Global Optimization* 20:133–154.

Tawarmalani M, Sahinidis NV (2013) *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*, volume 65 (Springer Science & Business Media).

Tsoukalas A, Mitsos A (2014) Multivariate McCormick relaxations. *Journal of Global Optimization* 59(2):633–662.

Vigerske S (2014) Towards MINLPLib 2.0 model instance collections .

Vigerske S, Gleixner A (2018) SCIP: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software* 33(3):563–593.

Wei J, Furman K, Duran M, Tawarmalani M, Sahinidis N (2005) Global optimization for nonconvex stochastic mixed-integer nonlinear programs. *International Federation of Operational Research Societies 2005 Triennial Conference* (Honolulu, Hawaii).

Westerlund T, Pettersson F (1995) An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering* 19:131–136.

Zhang Y, Sahinidis NV, Nohra C, Rong G (2020) Optimality-based domain reduction for inequality-constrained NLP and MINLP problems. *Journal of Global Optimization* 77:425–454.