Reformulation-Perspectification Technique for nonconvex optimization problems

In this paper, we propose a new global optimization approach for solving nonconvex optimization problems in which the nonconvex components are sums of products of convex functions. A broad class of nonconvex problems can be formulated in this way, such as concave minimization problems, problems with difference of convex functions in the objective and constraints, and fractional optimization problems. Our approach leverages two techniques: first, we introduce a new technique, called the Reformulation-Perspectification Technique (RPT), to obtain a convex approximation of the considered nonconvex continuous optimization problem. Next, we employ a spatial branch and bound scheme, utilizing RPT together with a modified eigenvector branching rule that extends Anstreicher (2022) to the case where $\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top$ may be indefinite, to obtain a global optimal solution. Numerical experiments on sum-of-max-linear-terms maximization, log-sum-exp maximization, linear multiplicative optimization, quadratically constrained quadratic optimization, and dike height optimization demonstrate the effectiveness of the proposed approach. Across these experiments, RPT-BB and RPT-SDP-BB are competitive with state-of-the-art global optimization solvers and are particularly effective on several harder instances.

Key words: Reformulation-Linearization Technique, perspective function, nonconvex optimization, conjugate function, branch and bound.

1. Introduction

We introduce a new global optimization approach for nonconvex optimization problems where the nonconvex elements are sums of products of convex functions. This formulation covers a wide range of optimization problems, including nonconvex quadratic optimization, mixed binary linear optimization, concave minimization, difference of convex programming, and fractional optimization.

For nonconvex quadratic optimization problems and mixed binary linear optimization problems, hierarchical convex approximations can be obtained from the Reformulation-Linearization Technique (RLT) (Sherali and Adams, 1999). RLT was introduced in Sherali and Adams (1990), and improved by many authors (Sturm and Zhang, 2003; Anstreicher, 2009, 2012, 2017; Bao et al., 2011; Yang and Burer, 2016; Jiang and Li, 2019). RLT is also applicable to mixed binary polynomial and to continuous nonconvex polynomial optimization problems (Sherali and Adams, 1999), and has been extended to mixed binary semi-infinite and convex optimization problems (Sherali and Adams, 2009). RLT consists of two steps: a reformulation step and a linearization step. The reformulation step generates redundant nonconvex constraints from pairwise multiplication of the existing linear or quadratic inequalities. The linearization step then substitutes each distinct product of variables by a continuous variable.

We also refer to Jiang and Li (2020) for an overview of RLT approximations for quadratic optimization problems.

We propose an extension of RLT, which we call the Reformulation-Perspectification Technique (RPT), to obtain a convex relaxation of the original nonconvex optimization problem. RPT consists of a reformulation and a perspectification step. Similarly to RLT, the reformulation step of RPT generates redundant nonconvex constraints from pairwise multiplication of the existing inequalities. Where in RLT only multiplications of linear or quadratic inequalities are considered, RPT also considers pairwise multiplications of not necessarily linear or quadratic convex inequalities, thereby handling more types of nonconvexity than RLT as well as potentially obtaining tighter approximations than RLT-based methods.

Moreover, in this paper we use a spatial branch and bound scheme, leveraging RPT and a modification of the eigenvector branching strategy proposed in Anstreicher (2022), to obtain a global optimal solution of the original nonconvex problem. A branch and bound algorithm was first introduced by Falk and Soland (1969), addressing optimization problems with continuous nonconvex separable objectives, and extended by Horst (1976) to non-separable functions, leveraging a different partitioning rule. In the context of nonconvex quadratically constrained quadratic problems (QCQPs), Al-Khayyal et al. (1995) develop a method for solving nonconvex QCQPs based on branch and bound, leveraging a linearization technique. Chen and Burer (2012) develop a branch and bound method, utilizing co-positive programming, addressing nonconvex quadratic problems over linear constraints. Anstreicher (2022) develops a spatial branching method termed eigenvector branching in order to strengthen semidefinite relaxations of problems with a nonconvex quadratic objective and/or constraints, leveraging the multiplication of linear with conic quadratic inequalities. In the context of nonlinear problems (NLPs) and mixed integer nonlinear problems (MINLPs), Ryoo and Sahinidis (1996) propose the branch and reduce algorithm, which is implemented in BARON (Sahinidis, 1996). The latter uses a branch and bound algorithm that iteratively solves convex relaxations of the initial problem and finds tighter variable bounds. BARON has been very successful so far and is in fact considered a state-of-the-art method for nonconvex optimization problems. Another state-of-the-art method that utilizes branch and bound, addressing nonconvex problems with an emphasis on integer problems, is the global optimization algorithm SCIP, developed by Achterberg (2009).

Although the idea of using branch and bound to obtain the global optimal solution of nonconvex optimization problems has already appeared in the literature, the novelty of RPT-BB lies in combining the new RPT relaxation with a modification of the eigenvector branching strategy of Anstreicher (2022). In particular, because $\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top$ need not be positive semidefinite in our setting, the branching rule is extended to also handle negative eigenvalues via an SOC-type child inequality.

For the problem of maximizing a twice continuously differentiable convex function over a convex compact feasible region, Selvi et al. (2022) develop an algorithm based on adjustable robust optimization, which can be shown to be a special case of our RPT approach. Additionally, Ben-Tal and Roos (2022) develop an algorithm called CoMax, which is based on gradient ascent. The latter is also applicable to integer optimization problems where the feasible set is a polytope. While both methods can find high-quality bounds on the optimal solution, neither can guarantee global optimality. Specifically, Selvi et al. (2022) is limited to problems with only one norm constraint or multiple linear constraints, while CoMax only obtains a lower bound for the optimal solution and imposes strict assumptions on the constraints.

Our main contributions can be summarized as follows:

1. We extend the existing RLT approach to a broader class of nonconvex optimization problems, namely optimization problems in which the nonconvex components are sums of products of convex functions. The proposed RPT approach can handle multiplication of constraints that are neither linear nor quadratic, and thereby potentially obtains tighter approximations than RLT. Moreover, it can also handle more types of nonconvexity than RLT.
2. We introduce RPT-BB, a new global optimization approach that combines the new RPT relaxation with a modification of the eigenvector branching strategy of Anstreicher (2022). Because $\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top$ need not be positive semidefinite in our setting, we extend the branching rule to also handle negative eigenvalues via an SOC-type child inequality. We show that the resulting RPT-BB approach converges to the optimal solution under Adaptation A in Section 6.
3. We provide several theoretical insights for our approach. We show that using epigraphical variables for the nonlinear convex components in the nonconvex objective and/or constraints, yields an RPT relaxation that is at least as tight as without the introduction of epigraphical variables. Further, we show that adding linear constraints that are redundant to existing linear constraints does not tighten the RPT relaxation, while adding linear constraints that are redundant to existing nonlinear constraints can be useful.
4. We demonstrate the effectiveness of the proposed RPT-based branch-and-bound methods through numerical experiments on five classes of problems: sum-of-max-linear-terms maximization, log-sum-exp maximization, linear multiplicative optimization, quadratically constrained quadratic optimization, and dike height optimization. The results show that RPT-BB and RPT-SDP-BB are competitive with state-of-the-art global optimization solvers and are particularly effective on several harder instances. In particular, for the sum-of-max-linear-terms

benchmark, our methods often outperform BARON on the more difficult instances and, in the nonconvex-feasible-region setting, RPT-BB is the only method that proves global optimality within the time limit for some instances; for the log-sum-exp benchmark, both RPT-BB and RPT-SDP-BB solve all instances to global optimality, while the biconjugate formulations remain difficult for BARON and SCIP; and for the dike height optimization problem, RPT-SDP-BB solves all tested instances at the root node in less than a second. In addition, we extensively study alternative branching implementations in Appendix A, which motivates the main-text choice of the Secant variant.

This paper is organized as follows. Section 2 introduces the generic nonconvex optimization problem we consider. Section 3 presents the RPT-BB approach. Section 4 illustrates the approach on a simple example. Section 5 discusses additional strengthening strategies. Section 6 provides the convergence analysis. Section 7 shows how several known convex reformulations and relaxations can also be obtained via RPT. Section 8 reports numerical experiments. Section 9 concludes.

Notation. We generally use bold faced characters such as $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ to represent vectors and matrices, respectively, a_i to denote the i-th element of the vector $\boldsymbol{a}$, $\boldsymbol{A}_i \in \mathbb{R}^m$ to denote the i-th column of matrix $\boldsymbol{A}$, and A_{ij} to denote the entry of $\boldsymbol{A}$ in the i-th row and j-th column, unless specified otherwise. The calligraphic letters $\mathcal{I}$, $\mathcal{J}$, $\mathcal{K}$, $\mathcal{L}$ and the corresponding capital Roman letters I, J, K, L are reserved for finite index sets and their respective cardinalities, *i.e.*, $\mathcal{I} = \{1,\dots,I\}$ etc. The subscript 0 for an index set indicates that the set additionally includes 0, *i.e.*, $\mathcal{I}_0 = \{0,\dots,I\}$ etc. Let $\mathbb{R}^{m\times n}$ denote the set of real $m\times n$ matrices, and $\mathbb{S}^n$ the set of real $n\times n$ symmetric matrices. We use $\mathrm{ri}(\mathcal{V})$ to denote the relative interior of a set $\mathcal{V} \subseteq \mathbb{R}^{n_\nu}$. The *domain* of a function $f : \mathbb{R}^{n_\nu} \to [-\infty,+\infty]$ is defined as $\mathrm{dom}(f) = \{\boldsymbol{\nu} \in \mathbb{R}^{n_\nu} \mid f(\boldsymbol{\nu}) < +\infty\}$. The function f is *proper* if $f(\boldsymbol{\nu}) > -\infty$ for all $\boldsymbol{\nu} \in \mathbb{R}^{n_\nu}$ and $f(\boldsymbol{\nu}) < +\infty$ for at least one $\boldsymbol{\nu} \in \mathbb{R}^{n_\nu}$, implying that $\mathrm{dom}(f) \neq \emptyset$. In addition, f is *closed* if f is lower semicontinuous and either $f(\boldsymbol{\nu}) > -\infty$ for all $\boldsymbol{\nu} \in \mathbb{R}^{n_\nu}$ or $f(\boldsymbol{\nu}) = -\infty$ for all $\boldsymbol{\nu} \in \mathbb{R}^{n_\nu}$. The *conjugate* of a function $f : \mathbb{R}^{n_\nu} \to [-\infty,+\infty]$ is the function $f^* : \mathbb{R}^{n_\nu} \to [-\infty,+\infty]$ defined through $f^*(\boldsymbol{w}) = \sup_{\boldsymbol{\nu}} \left\{\boldsymbol{\nu}^\top \boldsymbol{w} - f(\boldsymbol{\nu})\right\}$. The conjugate $(f^*)^*$ of f^* is called the *biconjugate* of f and is abbreviated as f^{**}. The *indicator function* $\delta_{\mathcal{V}} : \mathbb{R}^{n_\nu} \to [-\infty,+\infty]$ of a set $\mathcal{V} \subseteq \mathbb{R}^{n_\nu}$ is defined through $\delta_{\mathcal{V}}(\boldsymbol{\nu}) = 0$ if $\boldsymbol{\nu} \in \mathcal{V}$ and $\delta_{\mathcal{V}}(\boldsymbol{\nu}) = +\infty$ if $\boldsymbol{\nu} \notin \mathcal{V}$. The *support function* $\delta^*_{\mathcal{V}} : \mathbb{R}^{n_\nu} \to [-\infty,+\infty]$ of a set $\mathcal{V} \subseteq \mathbb{R}^{n_\nu}$ is defined through $\delta^*_{\mathcal{V}}(\boldsymbol{w}) = \sup_{\boldsymbol{\nu}\in\mathcal{V}} \{\boldsymbol{\nu}^\top \boldsymbol{w}\}$. The *perspective function* of a proper, closed and convex function $f : \mathbb{R}^{n_\nu} \to (-\infty,+\infty]$ is defined as $h(\boldsymbol{\nu},t) = tf(\boldsymbol{\nu}/t)$ if $t > 0$, and $h(\boldsymbol{\nu},0) = \delta^*_{\mathrm{dom}(f^*)}(\boldsymbol{\nu})$ for all $\boldsymbol{\nu} \in \mathbb{R}^{n_\nu}$. For ease of exposition, we use $tf(\boldsymbol{\nu}/t)$ to denote the perspective function $h(\boldsymbol{\nu},t)$ for the rest of this paper.

2. Generic problem formulation

We consider a generic nonconvex optimization problem of the following form:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_k(\boldsymbol{x}) \le 0, \quad k \in \mathcal{K}, \\ & \boldsymbol{x} \in \mathcal{X}, \end{aligned} \tag{1}$$

where $f_k : \mathbb{R}^{n_x} \to [-\infty, \infty]$ is a sum of convex times convex (SCC) function for all $k \in \mathcal{K}_0$, that is,

$$f_k(\boldsymbol{x}) = c_{0k}(\boldsymbol{x}) + \sum_{i \in \mathcal{I}_k} r_{ik}(\boldsymbol{x}) c_{ik}(\boldsymbol{x}),$$

and $c_{0k}, r_{ik}, c_{ik} : \mathbb{R}^{n_x} \to (-\infty, +\infty]$, are proper, closed and convex functions for every $i \in \mathcal{I}_k$, $k \in \mathcal{K}_0$. The set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is defined by:

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \boldsymbol{A}^\top \boldsymbol{x} \le \boldsymbol{b},\ \boldsymbol{P}^\top \boldsymbol{x} = \boldsymbol{s},\ \boldsymbol{h}(\boldsymbol{x}) \le \boldsymbol{0}\},$$

where $\boldsymbol{A} \in \mathbb{R}^{n_x \times m_1}$, $\boldsymbol{P} \in \mathbb{R}^{n_x \times m_2}$, $\boldsymbol{b} \in \mathbb{R}^{m_1}$, $\boldsymbol{s} \in \mathbb{R}^{m_2}$, $\boldsymbol{h}(\boldsymbol{x}) = [h_0(\boldsymbol{x})\ h_1(\boldsymbol{x})\ \cdots\ h_J(\boldsymbol{x})]^\top$, and $h_j : \mathbb{R}^{n_x} \to (-\infty, +\infty]$ are proper, closed and convex for every $j \in \mathcal{J}_0$. We make the following assumptions.

ASSUMPTION 1. *The set $\mathcal{X}$ is nonempty and compact.*

ASSUMPTION 2. *If r_{ik} and c_{ik} are both nonlinear, then $r_{ik}(\boldsymbol{x}) \ge 0$ and $c_{ik}(\boldsymbol{x}) \ge 0$ for all $\boldsymbol{x} \in \mathcal{X}$, for every $i \in \mathcal{I}$ and $k \in \mathcal{K}_0$. If r_{ik} is linear and c_{ik} is nonlinear, then $r_{ik}(\boldsymbol{x}) \ge 0$ for all $\boldsymbol{x} \in \mathcal{X}$, for every $i \in \mathcal{I}$ and $k \in \mathcal{K}_0$. If both r_{ik} and c_{ik} are linear, then we do not impose any assumption on these functions.*

Observe that we can reformulate an SCC function in the following way:

$$c_{0k}(\boldsymbol{x}) + \sum_{i \in \mathcal{I}_k} r_{ik}(\boldsymbol{x}) c_{ik}(\boldsymbol{x}) \le 0 \iff \begin{cases} c_{0k}(\boldsymbol{x}) + \sum_{i \in \mathcal{I}_k} \tau_{ik} c_{ik}(\boldsymbol{x}) \le 0, & \\ r_{ik}(\boldsymbol{x}) \le \tau_{ik}, & \text{if } r_{ik} \text{ and } c_{ik} \text{ are nonlinear,} \\ r_{ik}(\boldsymbol{x}) = \tau_{ik}, & \text{if } r_{ik} \text{ is linear.} \end{cases}$$

Hence in the remainder we can assume, without loss of generality, that the functions f_k in (1) are sum of linear times convex (SLC) functions for all $k \in \mathcal{K}_0$, that is,

$$f_k(\boldsymbol{x}) = c_{0k}(\boldsymbol{x}) + \sum_{i \in \mathcal{I}_k} \left(q_{ik} - \boldsymbol{d}_{ik}^\top \boldsymbol{x}\right) c_{ik}(\boldsymbol{x}), \tag{2}$$

and $q_{ik} \in \mathbb{R}$, $\boldsymbol{d}_{ik} \in \mathbb{R}^{n_x}$, and $c_{0k}, c_{ik} : \mathbb{R}^{n_x} \to (-\infty, +\infty]$ are proper, closed and convex for every $i \in \mathcal{I}_k$, and $k \in \mathcal{K}_0$.

We now present some examples of functions that are SLC or can be equivalently written as an SLC function.

Example 1 (Difference of convex functions). Consider the Difference of Convex (DC) function $f_k(\boldsymbol{x}) = c_{0k}(\boldsymbol{x}) - c_{1k}(\boldsymbol{x}) \leq 0$, where $c_{0k},\, c_{1k} : \mathbb{R}^{n_x} \to (-\infty, +\infty]$ are proper, closed and convex for some $k \in \mathcal{K}_0$. We can then reformulate the corresponding constraint function into an SLC function using the biconjugate reformulation (Rockafellar, 1970) and obtain

$$\begin{aligned} f_k(\boldsymbol{x}) \leq 0 \quad &\iff \quad \inf_{\boldsymbol{y} \in \mathrm{dom}(c_{1k}^*)} \{c_{0k}(\boldsymbol{x}) - \boldsymbol{x}^\top \boldsymbol{y} + c_{1k}^*(\boldsymbol{y})\} \leq 0 \\ &\iff \quad \exists \boldsymbol{y} \in \mathrm{dom}(c_{1k}^*) :\ c_{0k}(\boldsymbol{x}) - \boldsymbol{x}^\top \boldsymbol{y} + c_{1k}^*(\boldsymbol{y}) \leq 0, \end{aligned} \tag{3}$$

as long as the infimum is attained, since we can then remove the inf operator. In case the infimum is not attained we refer to Appendix B. We note that for many important classes of convex functions, their conjugates and domains are readily available from the literature. We summarize several of them in Table 1. We note that DC functions constitute an important class of SLC representable functions. For example, every twice continuously differentiable function has a DC decomposition (Hartman, 1959) and can therefore be written as an SLC function. Moreover, every concave function is a DC function, as we can take $c_{0k}(\boldsymbol{x}) = 0$.

Table 1 Example of functions $f(\cdot)$ and their corresponding conjugates. For the functions in #8, we assume that $\cap_i \mathrm{ri}(\mathrm{dom}(g_i)) \neq \emptyset$.

#	f	$\mathrm{dom}(f^*)$	f^*
1	$f(\boldsymbol{x}, \bar{x}) = \lVert\boldsymbol{x}\rVert_2 - \bar{x}$	$\{(\boldsymbol{y}, \bar{y}) : \lVert\boldsymbol{y}\rVert_2 \leq 1,\ \bar{y} = 1\}$	$f^*(\boldsymbol{y}, \bar{y}) = 0$
2	$f(x) = x\log(x)$	$\{y : y \in \mathbb{R}\}$	$f^*(y) = \exp(y-1)$
3	$f(x) = -\log(x)$	$\{y : y < 0\}$	$f^*(y) = -\log(-y) - 1$
4	$f(x) = \sqrt{x}$	$\{y : y < 0\}$	$f^*(y) = -\frac{1}{4y}$
5	$f(\boldsymbol{x}) = \max_i x_i$	$\{\boldsymbol{y} : \boldsymbol{y} \geq 0, \sum_i y_i = 1, \forall k\}$	$f^*(\boldsymbol{y}) = 0$
6	$f(\boldsymbol{x}) = \sum_i \max_{k \in \mathcal{K}_i} x_i$	$\{\{\boldsymbol{y}_i\}_i : \boldsymbol{y}_i \geq 0, \sum_{k \in \mathcal{K}_i} y_{ik} = 1, \forall i\}$	$f^*(\boldsymbol{y}) = 0$
7	$f(\boldsymbol{x}) = \log(\sum_i \exp(x_i))$	$\{\boldsymbol{y} : \boldsymbol{y} \geq 0, \sum_i y_i = 1\}$	$f^*(\boldsymbol{y}) = \sum_i y_i \log(y_i)$
8	$f(\boldsymbol{x}) = \sum_i g_i(\boldsymbol{x})$	$\{\{\boldsymbol{y}_i\}_i : \sum_i \boldsymbol{y}_i = \boldsymbol{y}, \boldsymbol{y}_i \in \mathrm{dom}(g_i^*), \forall i\}$	$f^*(\boldsymbol{y}) = \min_{\{\boldsymbol{y}_i\}_i} \sum_i g_i^*(\boldsymbol{y}_i)$

Example 2 (Fractional optimization). Consider the following fractional function

$$f(\boldsymbol{x}) = \sum_{i \in \mathcal{I}} \frac{c_i(\boldsymbol{x})}{r_i(\boldsymbol{x})},$$

where $c_i : \mathbb{R}^{n_x} \to \mathbb{R}_+$ is convex and $r_i : \mathbb{R}^{n_x} \to \mathbb{R}_{++}$ is concave for every $i \in \mathcal{I}$. Then f is not necessarily convex or concave. However, the function is SCC, since $1/r_i(\boldsymbol{x})$ is convex and nonnegative. □

Example 3 (Some examples of SLC functions). Table 2 lists several examples of SLC functions that are generally nonconvex. These functions are SLC-representable provided that Assumption 2 holds. Hence, our proposed approach can deal with Problem (1) containing (sums of) such nonconvex components. □

#	$f(\boldsymbol{x})$	$c_1(\boldsymbol{x})$	$(q-\boldsymbol{d}^\top\boldsymbol{x})$	Assumptions
1	$\sqrt{q-\boldsymbol{d}^\top\boldsymbol{x}}$	$(q-\boldsymbol{d}^\top\boldsymbol{x})^{-1/2}$	$q-\boldsymbol{d}^\top\boldsymbol{x}$	—
2	$(q-\boldsymbol{d}^\top\boldsymbol{x})^\theta$	$(q-\boldsymbol{d}^\top\boldsymbol{x})^{\theta-1}$	$q-\boldsymbol{d}^\top\boldsymbol{x}$	$\theta\in[0,1]$
3	$-(q-\boldsymbol{d}^\top\boldsymbol{x})^\theta$	$-(q-\boldsymbol{d}^\top\boldsymbol{x})^{\theta-1}$	$q-\boldsymbol{d}^\top\boldsymbol{x}$	$\theta\in[1,2]$
4	$-(q_1-\boldsymbol{d}_1^\top\boldsymbol{x})\ln(q_2-\boldsymbol{d}_2^\top\boldsymbol{x})$	$-\ln(q_2-\boldsymbol{d}_2^\top\boldsymbol{x})$	$q_1-\boldsymbol{d}_1^\top\boldsymbol{x}$	—
5	$(q-\boldsymbol{d}^\top\boldsymbol{x})\,\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}$	$\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}$	$q-\boldsymbol{d}^\top\boldsymbol{x}$	$\boldsymbol{Q}\succeq\boldsymbol{0}$

Table 2 Selected SLC-representable functions and their defining components.

3. Reformulation-Perspectification Technique and Branch and Bound

In this section, we describe our new approach, called RPT-BB, to obtain a global optimal solution of (1). Our approach comprises five steps:

Step 1: Preprocessing. Introduce epigraphical variables for every nonlinear convex component c_{0k}, $k\in\mathcal{K}_0$, in the nonconvex SLC functions.

Step 2: Reformulation and perspectification. Generate additional redundant nonconvex constraints from pairwise multiplication of the existing convex inequalities in (1). Next, convexify all nonconvex components in (1) and all nonconvex components in the additional generated constraints by reformulating them in their perspective form and subsequently linearizing all product terms.

Step 3 (Optional): SDP relaxation. Add an additional LMI inequality from the SDP relaxation of the linearization of all product terms.

Step 4: Obtaining upper bounds. Solve the convex RPT relaxation. From the solution of the RPT relaxation, construct a set of candidate solutions for (1), substitute these candidate solutions in Problem (1) and choose the best upper bound obtained.

Step 5: Branch and bound. Solve Problem (1) to optimality by means of a spatial branch and bound method. In the next sections, we describe each of these steps in more detail.

3.1. Preprocessing step

We introduce epigraphical variables for the convex component in the nonconvex SLC functions of (1), and from (2) we have

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{\tau}}\quad & \tau_0+\sum_{i\in\mathcal{I}_0}\left(q_{i0}-\boldsymbol{d}_{i0}^\top\boldsymbol{x}\right)c_{i0}(\boldsymbol{x})\\ \text{s.t.}\quad & \tau_k+\sum_{i\in\mathcal{I}_k}\left(q_{ik}-\boldsymbol{d}_{ik}^\top\boldsymbol{x}\right)c_{ik}(\boldsymbol{x})\le 0,\qquad k\in\mathcal{K},\\ & (\boldsymbol{x},\boldsymbol{\tau})\in\mathcal{T},\end{aligned}\tag{4}$$

where $\mathcal{T}=\{(\boldsymbol{x},\boldsymbol{\tau})\in\mathbb{R}^{n_x}\times\mathbb{R}^{K+1}\mid \boldsymbol{x}\in\mathcal{X},\ \boldsymbol{c}_0(\boldsymbol{x})\le\boldsymbol{\tau}\}$, and $\boldsymbol{c}_0(\boldsymbol{x})=[c_{00}(\boldsymbol{x})\ c_{01}(\boldsymbol{x})\ \cdots\ c_{0K}(\boldsymbol{x})]^\top\subseteq(-\infty,+\infty]^{K+1}$. As we will see later in Theorem 1, we can multiply these extra epigraphical constraints with the existing convex constraints to obtain a tighter convex relaxation.

3.2. Reformulation and perspectification

Now we are ready to explain the core idea of RPT. We first consider the univariate case. The intuition is as follows: Consider the convex function $c:\mathbb{R}\to\mathbb{R}$ and the nonconvex constraint set $\mathcal{X}=\{x\in\mathbb{R}_+:\ x\,c(x)\le 0\}$. The constraint set can be written as $\mathcal{X}=\left\{(x,x')\in\mathbb{R}_+\times\mathbb{R}_+:\ x\,c\left(\frac{x'}{x}\right)\le 0,\ x'=x^2\right\}$. Observe that the new set $\mathcal{X}$ is also nonconvex due to the constraint $x'=x^2$. However, observe that for $x\ge 0$, the function $h(x,x')=x\,c\left(\frac{x'}{x}\right)$ is the perspective function of $c(\cdot)$ and therefore it is jointly convex in x and x'. Thus, by either completely relaxing $x'=x^2$ and adding the hitherto redundant $x'\ge 0$, or by relaxing it as $x'\ge x^2$ to preserve convexity (and nonnegativity), one obtains a convex outer approximation of $\mathcal{X}$.

Remark 1. Observe that the RLT approach is a subcase of RPT as described above, assuming the function $c(\cdot)$ is linear. Since the perspective of a linear function is the function itself, the perspectification step is not needed in RLT. However, for nonlinear functions this is not the case and therefore the perspectification step is necessary.

In the general case, let f be an SLC function as given by (2), that satisfies Assumption 2. Then we can perspectify the generally nonconvex function f by first multiplying and dividing the argument of c_i by $(q_i-\boldsymbol{d}_i^\top\boldsymbol{x})$ for every $i\in\mathcal{I}$ to obtain the following equivalent reformulation of f:

$$f(\boldsymbol{x})=c_0(\boldsymbol{x})+\sum_{i\in\mathcal{I}}(q_i-\boldsymbol{d}_i^\top\boldsymbol{x})c_i\left(\frac{q_i\boldsymbol{x}-\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{d}_i}{q_i-\boldsymbol{d}_i^\top\boldsymbol{x}}\right).$$

Then, the quadratic terms $\boldsymbol{x}\boldsymbol{x}^\top$ in the argument of the reformulated f can be linearized by substituting $\boldsymbol{x}\boldsymbol{x}^\top$ with $\boldsymbol{X}\in\mathbb{S}^{n_x}$ to obtain the following sum of perspective functions:

$$\sum_{i\in\mathcal{I}}(q_i-\boldsymbol{d}_i^\top\boldsymbol{x})c_i\left(\frac{q_i\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}_i}{q_i-\boldsymbol{d}_i^\top\boldsymbol{x}}\right),\tag{5}$$

which is jointly convex in $(\boldsymbol{x},\boldsymbol{X})$ because c_i is convex if and only if its perspective is convex (Rockafellar, 1970). Observe that if Assumption 2 is not satisfied, i.e., $q_i-\boldsymbol{d}_i^\top\boldsymbol{x}\le 0$ for some $\boldsymbol{x}\in\mathcal{X}$ and $i\in\mathcal{I}$, then the above sum of perspective functions might not be convex. We obtain the following convex relaxation:

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{\tau},\boldsymbol{X}}\quad & \tau_0+\sum_{i\in\mathcal{I}_0}(q_{i0}-\boldsymbol{d}_{i0}^\top\boldsymbol{x})c_{i0}\left(\frac{q_{i0}\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}_{i0}}{q_{i0}-\boldsymbol{d}_{i0}^\top\boldsymbol{x}}\right)\\ \text{s.t.}\quad & \tau_k+\sum_{i\in\mathcal{I}_k}(q_{ik}-\boldsymbol{d}_{ik}^\top\boldsymbol{x})c_{ik}\left(\frac{q_{ik}\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}_{ik}}{q_{ik}-\boldsymbol{d}_{ik}^\top\boldsymbol{x}}\right)\le 0,\quad k\in\mathcal{K},\\ & (\boldsymbol{x},\boldsymbol{\tau})\in\mathcal{T}.\end{aligned}\tag{6}$$

By pairwise multiplying inequalities in the set $\mathcal{T}$, we can obtain additional redundant SLC constraints, which can then be convexified in a manner similar to what was described above. Once convexified, these SLC constraints are no longer redundant and actually serve as bounds on the newly introduced variables corresponding to the product terms. We can pairwise multiply the linear inequality constraints in the set $\mathcal{T}$, similar to the approach in RLT, to derive bounds on the newly introduced variables $\boldsymbol{X} \in \mathbb{S}^{n_x}$. However, with RPT, we can incrementally improve this approximation by also considering the pairwise multiplication of the linear and convex constraints in the set $\mathcal{T}$, followed by the pairwise multiplication of the convex inequalities in the set $\mathcal{T}$.

To be more precise, by considering the following cases of pairwise multiplication of the constraints in the set $\mathcal{T}$, we incrementally improve the convex relaxation of (1) derived from RPT:

Linear inequality $\times$ **Linear inequality.** This is well-known in RLT: we multiply the constraints $\boldsymbol{A}^\top \boldsymbol{x} \le \boldsymbol{b}$ of (1) with $\boldsymbol{A}^\top \boldsymbol{x} \le \boldsymbol{b}$, and obtain the redundant constraints:

$$\boldsymbol{b}\boldsymbol{x}^\top \boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{x}\boldsymbol{b}^\top \le \boldsymbol{A}^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{A} + \boldsymbol{b}\boldsymbol{b}^\top.$$

Since the (i,j)-th constraint is exactly the (j,i)-th constraint, we only consider the upper triangular of the matrix equations; so $m_1(m_1+1)/2$ constraints instead of m_1^2. Next, the nonlinear quadratic terms $\boldsymbol{x}\boldsymbol{x}^\top$ are linearized by substituting them with $\boldsymbol{X} \in \mathbb{S}^{n_x}$. We then obtain the following additional convex constraints:

$$\boldsymbol{b}\boldsymbol{x}^\top \boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{x}\boldsymbol{b}^\top \le \boldsymbol{A}^\top \boldsymbol{X} \boldsymbol{A} + \boldsymbol{b}\boldsymbol{b}^\top. \tag{7}$$

Moreover, we include the additional constraints

$$X_{ii} \ge 0, \quad i \in \{1, \dots, n_x\},$$

since $x_i^2 \ge 0$ for all $i \in \{1, \dots, n_x\}$.

Linear inequality $\times$ **Convex inequality.** By multiplying each ℓ-th linear inequality $\boldsymbol{A}_\ell^\top \boldsymbol{x} \le b_\ell$ of (1) with the convex constraints $\boldsymbol{c}_0(\boldsymbol{x}) \le \boldsymbol{\tau}$ and $\boldsymbol{h}(\boldsymbol{x}) \le \boldsymbol{0}$, we obtain $m_1(J+K+2)$ redundant SLC constraints of the form

$$(b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{c}_0(\boldsymbol{x}) \le (b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{\tau} \quad \text{and} \quad (b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}) \le \boldsymbol{0} \qquad \ell \in \{1, \dots, m_1\}.$$

Next, the redundant SLC constraints can be reformulated into:

$$(b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}) \le \boldsymbol{0} \quad \iff \quad (b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{h}\left(\frac{b_\ell \boldsymbol{x} - \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{A}_\ell}{b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x}}\right) \le \boldsymbol{0} \quad \text{and}$$

$$(b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{c}_0(\boldsymbol{x}) \le (b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{\tau} \quad \iff \quad (b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{c}_0\left(\frac{b_\ell \boldsymbol{x} - \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{A}_\ell}{b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x}}\right) \le (b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{\tau}.$$

Finally, the nonlinear quadratic terms $\boldsymbol{x}\boldsymbol{x}^\top$ and the bilinear terms $\boldsymbol{\tau}\boldsymbol{x}^\top$ are linearized by substituting them with $\boldsymbol{X} \in \mathbb{S}^{n_x}$ and $\boldsymbol{V} \in \mathbb{R}^{(K+1)\times n_x}$, to obtain the following additional convex constraints:

$$(b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{h}\left(\frac{b_\ell \boldsymbol{x} - \boldsymbol{X}\boldsymbol{A}_\ell}{b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x}}\right) \le \boldsymbol{0} \quad \text{and} \quad (b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{c}_0\left(\frac{b_\ell \boldsymbol{x} - \boldsymbol{X}\boldsymbol{A}_\ell}{b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x}}\right) \le b_\ell \boldsymbol{\tau} - \boldsymbol{V}\boldsymbol{A}_\ell.$$

Example 4 (Perspectified forms of selected SLC-representable functions). Table 3 complements Table 2 in Example 3 by showing the perspectified forms of the SLC-representable functions. □

#	$f(\boldsymbol{x})$	Perspectification	Assumptions
1	$\sqrt{q-\boldsymbol{d}^\top\boldsymbol{x}}$	$(q-\boldsymbol{d}^\top\boldsymbol{x})\sqrt{\frac{q-\boldsymbol{d}^\top\boldsymbol{x}}{q^2-2q\,\boldsymbol{d}^\top\boldsymbol{x}+\boldsymbol{d}^\top\boldsymbol{X}\boldsymbol{d}}}$	—
2	$(q-\boldsymbol{d}^\top\boldsymbol{x})^\theta$	$(q-\boldsymbol{d}^\top\boldsymbol{x})\left(\frac{q^2-2q\,\boldsymbol{d}^\top\boldsymbol{x}+\boldsymbol{d}^\top\boldsymbol{X}\boldsymbol{d}}{q-\boldsymbol{d}^\top\boldsymbol{x}}\right)^{\theta-1}$	$\theta\in[0,1]$
3	$-(q-\boldsymbol{d}^\top\boldsymbol{x})^\theta$	$-(q-\boldsymbol{d}^\top\boldsymbol{x})\left(\frac{q^2-2q\,\boldsymbol{d}^\top\boldsymbol{x}+\boldsymbol{d}^\top\boldsymbol{X}\boldsymbol{d}}{q-\boldsymbol{d}^\top\boldsymbol{x}}\right)^{\theta-1}$	$\theta\in[1,2]$
4	$-(q_1-\boldsymbol{d}_1^\top\boldsymbol{x})\ln(q_2-\boldsymbol{d}_2^\top\boldsymbol{x})$	$-(q_1-\boldsymbol{d}_1^\top\boldsymbol{x})\ln\left(\frac{q_1q_2-q_1\boldsymbol{d}_2^\top\boldsymbol{x}-q_2\boldsymbol{d}_1^\top\boldsymbol{x}+\boldsymbol{d}_1^\top\boldsymbol{X}\boldsymbol{d}_2}{q_1-\boldsymbol{d}_1^\top\boldsymbol{x}}\right)$	—
5	$(q-\boldsymbol{d}^\top\boldsymbol{x})\,\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}$	$\frac{(q\,\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d})^\top\boldsymbol{Q}(q\,\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d})}{q-\boldsymbol{d}^\top\boldsymbol{x}}$	$\boldsymbol{Q}\succeq\boldsymbol{0}$

Table 3 Perspectified forms of the SLC-representable functions in Table 2.

Linear equality $\times$ **Convex inequality.** When multiplying a linear equality constraint with a convex inequality constraint, the denominator and coefficient of the resulting perspective function are zero. Fortunately, all additional nonlinear constraints resulting from multiplying a linear equality constraint with a convex inequality constraint are redundant as long as we consider the pairwise multiplication of the linear equality constraints with all variables (see Lemma 1). For quadratic problems, a similar observation was first mentioned by Sherali and Adams (1999, Remark 8.1). Before we formally prove Lemma 1, we first define redundant constraints.

Definition 1 (Redundant equality constraints). An equality constraint $f(\boldsymbol{x})=0$, where $f:\mathbb{R}^{n_x}\to(-\infty,+\infty]$, is *redundant* to the nonempty set $\mathcal{X}\subseteq\mathbb{R}^{n_x}$, if $\mathcal{X}\subseteq\{\boldsymbol{x}\mid f(\boldsymbol{x})=0\}$.

Lemma 1. *Let $\boldsymbol{d}^\top\boldsymbol{x}=q$ be an equality constraint, where $\boldsymbol{d}\in\mathbb{R}^{n_x}$ and $q\in\mathbb{R}$. If the function $f:\mathbb{R}^{n_x}\to(-\infty,+\infty]$ is proper, closed and convex, then the constraint $(q-\boldsymbol{d}^\top\boldsymbol{x})f\left(\frac{q\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}}{q-\boldsymbol{d}^\top\boldsymbol{x}}\right)=0$ is redundant to $\left\{(\boldsymbol{x},\boldsymbol{X})\mid\boldsymbol{d}^\top\boldsymbol{x}=q,\ \boldsymbol{X}\boldsymbol{d}=q\boldsymbol{x}\right\}$.*

Proof. Fix $(\boldsymbol{x},\boldsymbol{X})$ such that $\boldsymbol{d}^\top\boldsymbol{x}=q$ and $\boldsymbol{X}\boldsymbol{d}=q\boldsymbol{x}$. Then $q-\boldsymbol{d}^\top\boldsymbol{x}=0$ and $q\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}=\boldsymbol{0}$. By the standard closure formula for the perspective of a proper, closed and convex function,

$$(q-\boldsymbol{d}^\top\boldsymbol{x})\,f\left(\frac{q\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}}{q-\boldsymbol{d}^\top\boldsymbol{x}}\right)=\delta^*_{\mathrm{dom}(f^*)}(\boldsymbol{0}).$$

Since f is proper, closed, and convex, its conjugate f^* is proper (Rockafellar, 1970, Thm. 12.2). By definition of properness, $\mathrm{dom}(f^*)\neq\emptyset$, hence

$$\delta^*_{\mathrm{dom}(f^*)}(\boldsymbol{0})=\sup_{\boldsymbol{y}\in\mathrm{dom}(f^*)}\langle\boldsymbol{0},\boldsymbol{y}\rangle=0.$$

Therefore, $(q-\boldsymbol{d}^\top\boldsymbol{x})f\left(\frac{q\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}}{q-\boldsymbol{d}^\top\boldsymbol{x}}\right)=0$ is satisfied by every $(\boldsymbol{x},\boldsymbol{X})$ in the set, i.e., it is redundant. □

Thanks to Lemma 1, it suffices to multiply each ℓ-th linear equality constraint $\boldsymbol{P}_\ell^\top\boldsymbol{x}=s_\ell$ with $\boldsymbol{x}$ and $\boldsymbol{\tau}$ respectively. We then obtain $m_2n_x+m_2(K+1)$ redundant SLC constraints of the form

$$(s_\ell-\boldsymbol{P}_\ell^\top\boldsymbol{x})\boldsymbol{x}=\boldsymbol{0} \quad \text{and} \quad (s_\ell-\boldsymbol{P}_\ell^\top\boldsymbol{x})\boldsymbol{\tau}=\boldsymbol{0} \qquad \ell\in\{1,\dots,m_2\}.$$

Finally, the nonlinear quadratic terms $\boldsymbol{x}\boldsymbol{x}^\top$ and the bilinear terms $\boldsymbol{\tau}\boldsymbol{x}^\top$ are linearized by substituting them with $\boldsymbol{X}\in\mathbb{S}^{n_x}$ and $\boldsymbol{V}\in\mathbb{R}^{(K+1)\times n_x}$. Including all additional constraints from pairwise multiplying the linear constraints with the linear and nonlinear convex constraints in Problem (6) we obtain the following convex relaxation:

$$\begin{array}{lll}
\min\limits_{\boldsymbol{x},\boldsymbol{\tau},\boldsymbol{X},\boldsymbol{V}} & \tau_0+\sum\limits_{i\in\mathcal{I}_0}(q_{i0}-\boldsymbol{d}_{i0}^\top\boldsymbol{x})c_{i0}\left(\dfrac{q_{i0}\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}_{i0}}{q_{i0}-\boldsymbol{d}_{i0}^\top\boldsymbol{x}}\right) & \\
\text{s.t.} & \tau_k+\sum\limits_{i\in\mathcal{I}_k}(q_{ik}-\boldsymbol{d}_{ik}^\top\boldsymbol{x})c_{ik}\left(\dfrac{q_{ik}\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}_{ik}}{q_{ik}-\boldsymbol{d}_{ik}^\top\boldsymbol{x}}\right)\le 0, & k\in\mathcal{K},\\
& \boldsymbol{b}\boldsymbol{x}^\top\boldsymbol{A}+\boldsymbol{A}^\top\boldsymbol{x}\boldsymbol{b}^\top\le\boldsymbol{A}^\top\boldsymbol{X}\boldsymbol{A}+\boldsymbol{b}\boldsymbol{b}^\top, & \\
& X_{ii}\ge 0, & i\in\{1,\dots,n_x\},\\
& (b_\ell-\boldsymbol{A}_\ell^\top\boldsymbol{x})\boldsymbol{h}\left(\dfrac{b_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{A}_\ell}{b_\ell-\boldsymbol{A}_\ell^\top\boldsymbol{x}}\right)\le\boldsymbol{0}, & \ell\in\{1,\dots,m_1\},\\
& (b_\ell-\boldsymbol{A}_\ell^\top\boldsymbol{x})\boldsymbol{c}_0\left(\dfrac{b_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{A}_\ell}{b_\ell-\boldsymbol{A}_\ell^\top\boldsymbol{x}}\right)\le b_\ell\boldsymbol{\tau}-\boldsymbol{V}\boldsymbol{A}_\ell, & \ell\in\{1,\dots,m_1\},\\
& s_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{P}_\ell=\boldsymbol{0}, & \ell\in\{1,\dots,m_2\},\\
& s_\ell\boldsymbol{\tau}-\boldsymbol{V}\boldsymbol{P}_\ell=\boldsymbol{0}, & \ell\in\{1,\dots,m_2\},\\
& (\boldsymbol{x},\boldsymbol{\tau})\in\mathcal{T}. &
\end{array} \tag{8}$$

Note that there are $m_1(J+1+\frac{m_1+1}{2})+(m_1+1)(K+1)+(m_2+1)n_x+m_2(K+1)$ additional constraints and $n_x^2+(n_x+1)(K+1)$ additional variables in (8) compared to (1).

The following theorem demonstrates that introducing epigraphical variables for the convex components of the SLC functions in (1) (as is done in the preprocessing step of RPT; see (4)), can result in a potentially tighter RPT relaxation.

THEOREM 1. *The convex relaxation (8) obtained from RPT with the preprocessing step is at least as tight as the relaxation obtained from RPT without the preprocessing step.*

Proof. Applying RPT without the preprocessing step yields the relaxation:

$$\begin{array}{lll}
\min\limits_{\boldsymbol{x},\boldsymbol{X}} & c_{00}(\boldsymbol{x})+\sum\limits_{i\in\mathcal{I}_0}(q_{i0}-\boldsymbol{d}_{i0}^\top\boldsymbol{x})c_{i0}\left(\dfrac{q_{i0}\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}_{i0}}{q_{i0}-\boldsymbol{d}_{i0}^\top\boldsymbol{x}}\right) & \\
\text{s.t.} & c_{0k}(\boldsymbol{x})+\sum\limits_{i\in\mathcal{I}_k}(q_{ik}-\boldsymbol{d}_{ik}^\top\boldsymbol{x})c_{ik}\left(\dfrac{q_{ik}\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}_{ik}}{q_{ik}-\boldsymbol{d}_{ik}^\top\boldsymbol{x}}\right)\le 0, & k\in\mathcal{K},\\
& \boldsymbol{b}\boldsymbol{x}^\top\boldsymbol{A}+\boldsymbol{A}^\top\boldsymbol{x}\boldsymbol{b}^\top\le\boldsymbol{A}^\top\boldsymbol{X}\boldsymbol{A}+\boldsymbol{b}\boldsymbol{b}^\top, & \\
& X_{ii}\ge 0, & i\in\{1,\dots,n_x\},\\
& (b_\ell-\boldsymbol{A}_\ell^\top\boldsymbol{x})\boldsymbol{h}\left(\dfrac{b_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{A}_\ell}{b_\ell-\boldsymbol{A}_\ell^\top\boldsymbol{x}}\right)\le\boldsymbol{0}, & \ell\in\{1,\dots,m_1\},\\
& s_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{P}_\ell=\boldsymbol{0}, & \ell\in\{1,\dots,m_2\},\\
& \boldsymbol{x}\in\mathcal{X}. &
\end{array} \tag{9}$$

Compared to (8), problem (9) omits the following two sets of constraints:

$$\begin{aligned} &(b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\, \boldsymbol{c}_0\!\left(\tfrac{b_\ell \boldsymbol{x} - \boldsymbol{X}\boldsymbol{A}_\ell}{b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x}}\right) \le b_\ell \boldsymbol{\tau} - \boldsymbol{V}\boldsymbol{A}_\ell, && \ell \in \{1,\dots,m_1\},\\ &s_\ell \boldsymbol{\tau} - \boldsymbol{V}\boldsymbol{P}_\ell = \boldsymbol{0}, && \ell \in \{1,\dots,m_2\}. \end{aligned} \tag{10}$$

Let (11) denote the problem obtained from (8) by dropping the two sets of constraints in (10):

$$\begin{array}{lll} \min\limits_{\boldsymbol{x},\boldsymbol{\tau},\boldsymbol{X},\boldsymbol{V}} & \tau_0 + \sum\limits_{i\in\mathcal{I}_0} (q_{i0} - \boldsymbol{d}_{i0}^\top \boldsymbol{x}) c_{i0}\left(\dfrac{q_{i0}\boldsymbol{x} - \boldsymbol{X}\boldsymbol{d}_{i0}}{q_{i0} - \boldsymbol{d}_{i0}^\top \boldsymbol{x}}\right) & \\ \text{s.t.} & \tau_k + \sum\limits_{i\in\mathcal{I}_k} (q_{ik} - \boldsymbol{d}_{ik}^\top \boldsymbol{x}) c_{ik}\left(\dfrac{q_{ik}\boldsymbol{x} - \boldsymbol{X}\boldsymbol{d}_{ik}}{q_{ik} - \boldsymbol{d}_{ik}^\top \boldsymbol{x}}\right) \le 0, & k \in \mathcal{K}, \\ & \boldsymbol{b}\boldsymbol{x}^\top \boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{x}\boldsymbol{b}^\top \le \boldsymbol{A}^\top \boldsymbol{X}\boldsymbol{A} + \boldsymbol{b}\boldsymbol{b}^\top, & \\ & X_{ii} \ge 0, & i \in \{1,\dots,n_x\}, \\ & (b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x})\boldsymbol{h}\left(\dfrac{b_\ell \boldsymbol{x} - \boldsymbol{X}\boldsymbol{A}_\ell}{b_\ell - \boldsymbol{A}_\ell^\top \boldsymbol{x}}\right) \le \boldsymbol{0}, & \ell \in \{1,\dots,m_1\}, \\ & s_\ell \boldsymbol{x} - \boldsymbol{X}\boldsymbol{P}_\ell = \boldsymbol{0}, & \ell \in \{1,\dots,m_2\}, \\ & c_{0k}(\boldsymbol{x}) \le \tau_k, & k \in \mathcal{K}_0, \\ & \boldsymbol{x} \in \mathcal{X}. & \end{array} \tag{11}$$

Since dropping constraints can only enlarge the feasible region, it follows that $\min(8) \ge \min(11)$. Next, observe that after removing (10), the variable $\boldsymbol{V}$ no longer appears in any constraint and can therefore be eliminated without affecting feasibility or the minimum. Moreover, in (11) the variable $\boldsymbol{\tau}$ appears only through epigraph constraints, and the objective is nondecreasing in $\boldsymbol{\tau}$; hence τ_k can be eliminated by setting it to $c_{0k}(\boldsymbol{x})$ for every $k \in \mathcal{K}_0$. Observe that, eliminating $\boldsymbol{\tau}$ from (11) recovers exactly (9). Hence, $\min(11) = \min(9)$, which implies that $\min(8) \ge \min(9)$. Therefore, (8) is at least as tight as (9). □

The following example shows that introducing epigraphical variables for the nonlinear convex components in the preprocessing step can indeed tighten the RPT relaxation.

Example 5 (Effectiveness of preprocessing). Consider the following nonconvex optimization problem:

$$\begin{array}{ll} \max\limits_{x} & x_1 x_2 - \frac{(x_2-2)^2}{4} \\ \text{s.t.} & 0 \le x_1 \le 1. \end{array} \tag{12}$$

In the following we compare two convex relaxations of (12) obtained from (i) applying RPT without the epigraphical reformulation, and (ii) applying RPT with the epigraphical reformulation, respectively,

$$\begin{array}{lll} \max_{\boldsymbol{x},\boldsymbol{X}} & X_{12} - \frac{(x_2-2)^2}{4} \\ \text{s.t.} & 0 \le x_1 \le 1, \\ & X_{11} - 2x_1 + 1 \ge 0, \\ & x_1 - X_{11} \ge 0, \\ & X_{11} \ge 0, \end{array} \quad \text{and} \quad \begin{array}{ll} \max_{\boldsymbol{x},\boldsymbol{X},\tau,v} & X_{12} - \tau \\ \text{s.t.} & 0 \le x_1 \le 1, \\ & X_{11} - 2x_1 + 1 \ge 0, \\ & x_1 - X_{11} \ge 0, \\ & X_{11} \ge 0, \\ & \frac{(x_2-2)^2}{4} \le \tau, \\ & \frac{(X_{12}-2x_1)^2}{4x_1} \le v, \\ & \frac{(x_2 - 2 - X_{12} + 2x_1)^2}{4 - 4x_1} \le \tau - v. \end{array}$$

Note that the maximum objective value of the convex relaxation without the epigraphical reformulation is ∞, where $x_1^\star \in [0,1]$, $x_2^\star = 2$, $X_{11}^\star \in [\max(2x_1^\star - 1, 0), x_1^\star]$ and $X_{12}^\star = \infty$. In contrast, the objective value of the epigraphical reformulation is 3, with the optimal solution $(x_1^\star, x_2^\star, X_{11}^\star, X_{12}^\star, \tau^\star, v^\star) = (1,4,1,4,1,1)$. □

However, as demonstrated in the following lemma, introducing epigraphical variables for the linear components does not enhance the convex relaxation obtained from RPT.

Lemma 2. *Introducing epigraphical variables for the linear components in the preprocessing step of RPT does not result in a tighter convex relaxation.*

Proof. Suppose $c_{0k} : \mathbb{R}^{n_x} \to \mathbb{R}$ is a linear function in (1) for some $k \in \mathcal{K}_0$. In the preprocessing step of RPT, one can introduce an epigraphical variable τ_k for the linear component c_{0k} and consider the linear equality $\tau_k = c_{0k}(\boldsymbol{x})$ in the constraints. From Lemma 1, we know that it suffices to consider the multiplication of this constraint by $\boldsymbol{x}$ and $\boldsymbol{\tau}$ to obtain the convex relaxation from RPT. Specifically, from the reformulation and perspectification step of RPT, we obtain:

$$V_{ki} = x_i c_{0k}\left(\frac{\boldsymbol{X}_i}{x_i}\right), \quad i \in \{1,\ldots,n_x\}, \quad \text{and} \quad T_{kk'} = \tau_{k'} c_{0k}\left(\frac{(\boldsymbol{V}^\top)_{k'}}{\tau_{k'}}\right), \quad k' \in \mathcal{K}_0,$$

where $\boldsymbol{V}_k$ and $\boldsymbol{T}_k$ are variables introduced to linearize the product terms $\tau_k \boldsymbol{x}$ and $\tau_k \boldsymbol{\tau}$, respectively. Since $\boldsymbol{V}_k$ and $\boldsymbol{T}_k$ do not appear anywhere else but in these linear equalities in the convex relaxation from RPT, these equalities exactly determine $\boldsymbol{V}_k$ and $\boldsymbol{T}_k$ and are therefore redundant. □

Convex inequality $\times$ Convex inequality. Just multiplying a nonlinear convex constraint $h_j(\boldsymbol{x}) \le 0$ with another nonlinear convex constraint $h_{j'}(\boldsymbol{x}) \le 0$ results in a constraint $-h_j(\boldsymbol{x})h_{j'}(\boldsymbol{x}) \le 0$ for which the constraint function is not an SCC function, since in this case $-h_j(\boldsymbol{x})$ is concave instead of convex. However, sometimes rewriting the constraints, and then multiplying the left-hand sides and right-hand sides of the constraints yields convexifiable constraints. Consider for example the following two exponential constraints:

$$\exp(x_1) \le x_2 \quad \text{and} \quad \exp(x_3) \le x_4.$$

We can then multiply the left-hand sides and the right-hand sides, and multiply the right-hand side of each constraint with the other exponential constraint to obtain the following convexified constraints:

$$\begin{cases} \exp(x_1 + x_3) \leq X_{24}, \\ x_4 \exp\left(\dfrac{X_{14}}{x_4}\right) \leq X_{24}, \\ x_2 \exp\left(\dfrac{X_{23}}{x_2}\right) \leq X_{24}. \end{cases}$$

Also, several ways of obtaining a convexifiable constraint from pairwise multiplication of conic quadratic constraints are readily available in the literature (Yang and Burer, 2016; Anstreicher, 2017; Jiang and Li, 2019).

A related mechanism arises when a nonlinear convex inequality is replaced by a redundant linear support (e.g., a supporting hyperplane) and then combined with a second nonlinear convex inequality. Although such linear supports do not change the feasible region, they can generate additional valid inequalities after RPT, thereby tightening the relaxation. An analogous observation has already been made in the literature for RLT; see, for example, the SOC–RLT construction in Burer and Anstreicher (2013).

3.3. Additional SDP relaxation

In order to further tighten the convex relaxation, effective SDP cuts can be considered. In the perspectification step of RPT, the nonconvex quadratic terms $\boldsymbol{x}\boldsymbol{x}^\top$ are linearized by a symmetric matrix $\boldsymbol{X}$. Such a linearization based relaxation for the nonconvex quadratic equality $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$ may be significantly improved by the SDP relaxation $\boldsymbol{X} \succeq \boldsymbol{x}\boldsymbol{x}^\top$, which can be equivalently reformulated as an LMI by using Schur complement (Boyd and Vandenberghe, 2004):

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{x} \\ \boldsymbol{x}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0}. \tag{13}$$

Because we also have epigraphical constraints, we can consider including the following LMI:

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{V}^\top & \boldsymbol{x} \\ \boldsymbol{V} & \boldsymbol{T} & \boldsymbol{\tau} \\ \boldsymbol{x}^\top & \boldsymbol{\tau}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0},$$

where $\boldsymbol{T} \in \mathbb{S}^{K+1}$ denotes the matrix that substitutes the quadratic terms $\boldsymbol{\tau}\boldsymbol{\tau}^\top$. Although including the LMI might tighten the convex RPT relaxation, it can significantly increase computation time. Hence, this step is optional. Observe that if the above LMI is included, the additional constraints $X_{ii} \geq 0,\ i \in \{1, \ldots, n_x\}$, are redundant to the LMI.

3.4. Obtaining upper bounds

Let $(\boldsymbol{x}^*, \boldsymbol{\tau}^*, \boldsymbol{X}^*, \boldsymbol{V}^*)$ be the solution of the convex RPT relaxation (8). We propose to construct the set $\mathcal{X}' = \{\boldsymbol{x}^*, \boldsymbol{x}_1^X, \dots, \boldsymbol{x}_{n_x}^X, \boldsymbol{x}_1^V, \dots, \boldsymbol{x}_{n_\tau}^V\}$ of candidate solutions for (1), where

$$\boldsymbol{x}_i^X = \begin{cases} \boldsymbol{x}^* & \text{if } x_i^* = 0, \\ \frac{\boldsymbol{X}_i^*}{x_i^*} & \text{otherwise}, \end{cases} \quad \text{and} \quad \boldsymbol{x}_k^V = \begin{cases} \boldsymbol{x}^* & \text{if } \tau_k^* = 0, \\ \frac{(\boldsymbol{V}^{*\top})_k}{\tau_k^*} & \text{otherwise}, \end{cases} \quad \text{for all } i \in \{1, \dots, n_x\},\ k \in \mathcal{K}_0.$$

Note that by definition $\boldsymbol{x}^* \in \mathcal{X}$ is always satisfied.

LEMMA 3. *If $\mathcal{X}$ explicitly contains the nonnegativity constraints $\boldsymbol{x} \geq \boldsymbol{0}$, then $\mathcal{X}' \subset \mathcal{X}$.*

Proof. Since $\mathcal{X}$ explicitly contains the nonnegativity constraints $\boldsymbol{x} \geq \boldsymbol{0}$, by construction, if $x_i^* \neq 0$, it follows from $x_i^* \geq 0$ and $\boldsymbol{x}_i^X = \frac{\boldsymbol{X}_i^*}{x_i^*}$ that

$$\begin{cases} \boldsymbol{A}^\top \boldsymbol{X}_i^* \leq x_i^* \boldsymbol{b} \\ \boldsymbol{P}^\top \boldsymbol{X}_i^* = x_i^* \boldsymbol{s} \\ x_i^* \boldsymbol{h}\left(\frac{\boldsymbol{X}_i^*}{x_i^*}\right) \leq 0 \end{cases} \implies \begin{cases} \boldsymbol{A}^\top \frac{\boldsymbol{X}_i^*}{x_i^*} \leq \boldsymbol{b} \\ \boldsymbol{P}^\top \frac{\boldsymbol{X}_i^*}{x_i^*} = \boldsymbol{s} \\ \boldsymbol{h}\left(\frac{\boldsymbol{X}_i^*}{x_i^*}\right) \leq 0. \end{cases}$$

Otherwise, if $x_i^* = 0$, we have $\boldsymbol{x}_i^X = \boldsymbol{x}^*$ by definition, hence $\boldsymbol{x}_i^X \in \mathcal{X}$. Therefore, $\boldsymbol{x}_i^X \in \mathcal{X}$ for every $i \in \{1, \dots, n_x\}$. Analogously, one can show that $\boldsymbol{x}_k^V \in \mathcal{X}$ for every $k \in \mathcal{K}_0$. □

In cases where all constraints are convex (i.e., $\mathcal{K} = \emptyset$), substituting the candidate solutions into the original Problem (1) yields upper bounds corresponding to each candidate solution, allowing us to choose the best upper bound obtained. However, if $\boldsymbol{x}$ is not assumed to be nonnegative, or if we also have nonconvexity in the constraints, then the candidate solutions in $\mathcal{X}'$ obtained from the RPT relaxation may not be feasible for Problem (1). In such scenarios, to determine the best upper bound, we propose considering only those candidate solutions that are feasible for Problem (1).

Observe that if we have nonconvexities in the constraints, it is possible that none of the candidate solutions is feasible. Furthermore, note that if only some of the x_i, $i \in \{1, \dots, n_x\}$, are assumed to be nonnegative, it is possible to obtain an upper bound by considering only those candidate solutions $\frac{\boldsymbol{X}_i^*}{x_i^*}$ where the indices i correspond to positive x_i^*. This can be proved similarly to the proof of Lemma 3.

The obtained feasible solutions could also be used as warm starts for existing algorithms, to improve the upper bound. Namely, we can use a local optimization algorithm, such as the Ipopt solver (Waechter et al., 2009), initialized at the candidate feasible solution, to obtain a local optimum. We then replace the candidate solution in $\mathcal{X}'$ by the obtained local optimum. Note that we can also initialize it from an infeasible solution, and if the solver finds a feasible solution, we also add the solution to $\mathcal{X}''$. Moreover, for the problem of minimizing a concave or a difference of convex function, using the biconjugate reformulation, the problem can be formulated as a disjoint bilinear optimization problem, where the bilinear function is in fact an SLC function (see Example 1). Hence, we can leverage the mountain climbing algorithm by (Tao and An, 1997), to find a local optimum of (1), see Appendix C.

3.5. Spatial branch and bound method

We solve Problem (1) to global optimality via a spatial branch and bound scheme. Each node N_j of the search tree is associated with a relaxation region $\mathcal{X}_j \subseteq \mathbb{R}^{n_x} \times \mathbb{S}^{n_x}$ that collects the constraints on $(\boldsymbol{x}, \boldsymbol{X})$ inherited along the path to N_j (including $\boldsymbol{x} \in \mathcal{X}$). At node N_j, we solve the convex RPT relaxation (8) with the additional restriction $(\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j$; this yields a valid lower bound. A feasible solution (and hence an upper bound) is obtained using the techniques in Section 3.4. The algorithm terminates when the global optimality gap $(\text{Ub} - \text{Lb})$ is at most δ, a user-specified optimality tolerance.

Before formally defining the branching strategy, we make the following observation. Problem (8) is a relaxation of Problem (1) obtained by omitting the rank-one condition $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$. Any feasible $\boldsymbol{x}$ of (1) can be lifted to a feasible solution of (8) by setting $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$ (and, when epigraph variables are present, choosing $\boldsymbol{\tau}$ accordingly and setting $\boldsymbol{V} = \boldsymbol{\tau}\boldsymbol{x}^\top$). Moreover,

$$\left\{(\boldsymbol{x}, \boldsymbol{X}) \mid \boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top\right\} = \left\{(\boldsymbol{x}, \boldsymbol{X}) \;\middle|\; \forall \boldsymbol{\nu} \in \mathbb{R}^{n_x} : (\boldsymbol{\nu}^\top \boldsymbol{x})^2 \leq \boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu}, \;\; \boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu} \leq (\boldsymbol{\nu}^\top \boldsymbol{x})^2 \right\}. \tag{14}$$

Thus, omitting the nonconvex constraints in (14) yields the convex relaxation (8), which can be solved efficiently, but whose solution is often infeasible for Problem (1).

To address violations of (14) in solutions of (8), we use the following eigenvector branching strategy. Let $(\boldsymbol{x}^*, \boldsymbol{\tau}^*, \boldsymbol{X}^*, \boldsymbol{V}^*)$ be an optimal solution of the RPT relaxation at node N_j, and let $(\lambda, \boldsymbol{\nu})$ denote an eigenpair of the symmetric matrix $\boldsymbol{X}^* - \boldsymbol{x}^*(\boldsymbol{x}^*)^\top$, where $|\lambda|$ is maximal and $\boldsymbol{\nu}$ is the corresponding unit eigenvector. Set $l_j = \boldsymbol{\nu}^\top \boldsymbol{x}^*$ and branch on the hyperplane $\boldsymbol{\nu}^\top \boldsymbol{x} = l_j$. The rank-one condition requires $\boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu} = (\boldsymbol{\nu}^\top \boldsymbol{x})^2$, while at the relaxation solution we have

$$\boldsymbol{\nu}^\top \boldsymbol{X}^* \boldsymbol{\nu} - (\boldsymbol{\nu}^\top \boldsymbol{x}^*)^2 = \lambda, \tag{15}$$

so $|\lambda|$ measures the rank-one violation in direction $\boldsymbol{\nu}$. Hence, eigenvector branching adaptively selects a direction of maximal rank-one violation (Anstreicher, 2022). We now define the branching rule according to the sign of λ.

Case $\lambda < 0$. We add the convex inequality $(\boldsymbol{\nu}^\top \boldsymbol{x})^2 \leq \boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu}$ in each child and define

$$\begin{aligned}
\mathcal{X}_{j_1} &= \left\{(\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j \;\middle|\; \boldsymbol{\nu}^\top \boldsymbol{x} \leq l_j, \;\; (\boldsymbol{\nu}^\top \boldsymbol{x})^2 \leq \boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu} \right\}, \\
\mathcal{X}_{j_2} &= \left\{(\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j \;\middle|\; \boldsymbol{\nu}^\top \boldsymbol{x} \geq l_j, \;\; (\boldsymbol{\nu}^\top \boldsymbol{x})^2 \leq \boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu} \right\}.
\end{aligned}$$

The current solution $(\boldsymbol{x}^*, \boldsymbol{X}^*)$ is infeasible in both children because $(\boldsymbol{\nu}^\top \boldsymbol{x}^*)^2 - \boldsymbol{\nu}^\top \boldsymbol{X}^* \boldsymbol{\nu} = -\lambda > 0$ violates $(\boldsymbol{\nu}^\top \boldsymbol{x})^2 \leq \boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu}$.

Case $\lambda > 0$. We would like to impose $\boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu} \le (\boldsymbol{\nu}^\top \boldsymbol{x})^2$ in each child, but this inequality is nonconvex. We therefore replace it by secant inequalities, following Saxena et al. (2010) and Anstreicher (2022). Define

$$\underline{l}_j := \min\{\boldsymbol{\nu}^\top \boldsymbol{x} : (\boldsymbol{x},\boldsymbol{X}) \in \mathcal{X}_j\}, \qquad \bar{l}_j := \max\{\boldsymbol{\nu}^\top \boldsymbol{x} : (\boldsymbol{x},\boldsymbol{X}) \in \mathcal{X}_j\}.$$

The two convex child regions are then

$$\mathcal{X}_{j_1} = \left\{(\boldsymbol{x},\boldsymbol{X}) \in \mathcal{X}_j \;\middle|\; \boldsymbol{\nu}^\top \boldsymbol{x} \le l_j, \;\; \boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu} \le (\underline{l}_j + l_j)\,\boldsymbol{\nu}^\top \boldsymbol{x} - \underline{l}_j\, l_j \right\},$$
$$\mathcal{X}_{j_2} = \left\{(\boldsymbol{x},\boldsymbol{X}) \in \mathcal{X}_j \;\middle|\; \boldsymbol{\nu}^\top \boldsymbol{x} \ge l_j, \;\; \boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu} \le (l_j + \bar{l}_j)\,\boldsymbol{\nu}^\top \boldsymbol{x} - l_j\, \bar{l}_j \right\}.$$

The current solution $(\boldsymbol{x}^*, \boldsymbol{X}^*)$ is again infeasible in both child relaxations because $\boldsymbol{\nu}^\top \boldsymbol{X}^*\boldsymbol{\nu} = (\boldsymbol{\nu}^\top \boldsymbol{x}^*)^2 + \lambda$, whereas both secant right-hand sides evaluate to $l_j^2 = (\boldsymbol{\nu}^\top \boldsymbol{x}^*)^2$ at $\boldsymbol{\nu}^\top \boldsymbol{x}^* = l_j$. Moreover, these secant inequalities are valid for all rank-one solutions: if $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$, then $\boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu} = (\boldsymbol{\nu}^\top \boldsymbol{x})^2$, and on $[\underline{l}_j, l_j]$ and $[l_j, \bar{l}_j]$ the function $(\boldsymbol{\nu}^\top \boldsymbol{x})^2$ lies below its secant. Since the split is exhaustive, any feasible (in particular, globally optimal) rank-one solution remains feasible in at least one child.

Branching at node N_j therefore does not exclude any feasible solution of Problem (1) that is consistent with N_j. Indeed, let $\boldsymbol{x}$ be feasible for Problem (1) and suppose $(\boldsymbol{x}, \boldsymbol{x}\boldsymbol{x}^\top) \in \mathcal{X}_j$. Then $\boldsymbol{x}$ can be lifted to a feasible solution of the node relaxation by setting $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$ (and, when epigraph variables are present, choosing $\boldsymbol{\tau}$ accordingly and setting $\boldsymbol{V} = \boldsymbol{\tau}\boldsymbol{x}^\top$). By the case analysis above, the child inequalities are valid for such rank-one lifted solutions, and since the split $\boldsymbol{\nu}^\top \boldsymbol{x} \le l_j$ or $\boldsymbol{\nu}^\top \boldsymbol{x} \ge l_j$ is exhaustive, the lifted solution remains feasible in at least one child. Consequently, branching only on $(\boldsymbol{x}, \boldsymbol{X})$ is sufficient for correctness: it does not exclude any feasible (in particular, globally optimal) solution of Problem (1). The auxiliary variables $\boldsymbol{\tau}$ and $\boldsymbol{V}$ need not be branched on and are re-optimized in each child relaxation.

If the semidefinite constraint $\begin{pmatrix} \boldsymbol{X} & \boldsymbol{x} \\ \boldsymbol{x}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0}$ is enforced, then $\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top \succeq \boldsymbol{0}$ holds for all feasible points of the relaxation, and hence $\lambda < 0$ cannot occur. Without this constraint, $\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top$ may be indefinite, and either sign of λ can arise; the branching rule above covers both cases. We treat $|\lambda| \le \delta_{\text{eig}}$ (for a prescribed tolerance $\delta_{\text{eig}} > 0$) as indicating no significant rank-one violation in the selected direction. In that event, eigenvector branching provides little refinement, so we apply a coordinate split as a fallback spatial refinement step (see Section 6 for more details).

After branching, we solve the RPT relaxation over each child region (optimizing over $(\boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{X}, \boldsymbol{V})$ subject to $(\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_{j_k}$) to obtain lower bounds and update the global upper bound using Section 3.4. The branching rule is summarized in Algorithm 1. To construct the RPT relaxation at a node, we perform constraint multiplications only for inequalities involving $\boldsymbol{x}$ alone, including the branching inequality $\boldsymbol{\nu}^\top \boldsymbol{x} \le l_j$ or $\boldsymbol{\nu}^\top \boldsymbol{x} \ge l_j$. We do not use constraints involving $\boldsymbol{X}$ (e.g., the secant inequalities or $(\boldsymbol{\nu}^\top \boldsymbol{x})^2 \le \boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu})$ for constraint multiplications, as doing so would introduce products such as $x_i X_{k\ell}$ (i.e., third-order terms); instead, we keep these constraints in their original form. We prune a node whenever its lower bound exceeds the current best upper bound and select the next node using best-bound selection (i.e., the active node with the smallest lower bound). Algorithm 2 summarizes the overall procedure.

REMARK 2. Since the eigenvector $\boldsymbol{\nu}$ corresponding to the eigenvalue with the largest absolute value is typically dense, the resulting branching constraints are dense as well. Dense constraints may increase per-node solve time relative to coordinate branching; however, in our experiments we did not observe a consistent or dominant runtime impact attributable to this density. Moreover, a dense $\boldsymbol{\nu}$ couples many components of $\boldsymbol{x}$, which can help tighten the relaxation when RPT is applied.

Algorithm 1 Eigenvector branching strategy

Input: $\mathcal{X}_j$, δ_{eig}.

Output: $(\mathcal{X}_{j_1}, \mathcal{X}_{j_2})$.

Solve the RPT relaxation over $\mathcal{X}_j$ and obtain $(\boldsymbol{x}^*, \boldsymbol{X}^*)$.
Compute an eigenpair $(\lambda, \boldsymbol{\nu})$ of $\boldsymbol{X}^* - \boldsymbol{x}^*(\boldsymbol{x}^*)^\top$ such that $|\lambda|$ is maximized and $\|\boldsymbol{\nu}\|_2 = 1$.
if $|\lambda| \le \delta_{\text{eig}}$ **then**
  (Fallback) Apply an exhaustive spatial refinement rule to $\mathcal{X}_j$ (e.g., a coordinate split; see Section 6) and return the resulting two child regions $\mathcal{X}_{j_1}$ and $\mathcal{X}_{j_2}$.
else if $\lambda < 0$ **then**
  Set $l_j := \boldsymbol{\nu}^\top \boldsymbol{x}^*$.
  $\mathcal{X}_{j_1} \leftarrow \{(\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j \mid \boldsymbol{\nu}^\top \boldsymbol{x} \le l_j,\ (\boldsymbol{\nu}^\top \boldsymbol{x})^2 \le \boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu}\}$.
  $\mathcal{X}_{j_2} \leftarrow \{(\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j \mid \boldsymbol{\nu}^\top \boldsymbol{x} \ge l_j,\ (\boldsymbol{\nu}^\top \boldsymbol{x})^2 \le \boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu}\}$.
else
  Set $l_j := \boldsymbol{\nu}^\top \boldsymbol{x}^*$.
  $\underline{l}_j \leftarrow \min\{\boldsymbol{\nu}^\top \boldsymbol{x} \mid (\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j\}$; $\bar{l}_j \leftarrow \max\{\boldsymbol{\nu}^\top \boldsymbol{x} \mid (\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j\}$.
  $\mathcal{X}_{j_1} \leftarrow \{(\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j \mid \boldsymbol{\nu}^\top \boldsymbol{x} \le l_j,\ \boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu} \le (\underline{l}_j + l_j)\boldsymbol{\nu}^\top \boldsymbol{x} - \underline{l}_j l_j\}$.
  $\mathcal{X}_{j_2} \leftarrow \{(\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_j \mid \boldsymbol{\nu}^\top \boldsymbol{x} \ge l_j,\ \boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu} \le (l_j + \bar{l}_j)\boldsymbol{\nu}^\top \boldsymbol{x} - l_j \bar{l}_j\}$.
end if
return $\mathcal{X}_{j_1}, \mathcal{X}_{j_2}$.

Algorithm 2 Branch and bound via RPT

Input: $(N_0, \mathcal{X}_0, \text{Lb}_0, \text{Ub}_0, \delta)$.

Output: $(\boldsymbol{x}^*, \text{Lb}, \text{Ub})$.

$\text{Lb} \leftarrow \text{Lb}_0$; $\quad \text{Ub} \leftarrow \text{Ub}_0$.
ACTIVE $\leftarrow \{N_0\}$.
while $\text{Ub} - \text{Lb} > \delta$ **do**
  $N_j \leftarrow \arg\min_{N_i \in \text{ACTIVE}} \text{Lb}_i$.
  Partition node N_j by constructing $\mathcal{X}_{j_1}, \mathcal{X}_{j_2}$ from $\mathcal{X}_j$ via Algorithm 1.
  for $k = 1, 2$ **do**
    $i \leftarrow j_k$.
    Solve the RPT relaxation over $\mathcal{X}_i$ at node N_i and obtain Lb_i and Ub_i.
    if $\text{Ub}_i < \text{Ub}$ **then**
      $\text{Ub} \leftarrow \text{Ub}_i$ and update the incumbent $\boldsymbol{x}^*$.
    end if
  end for
  for $k = 1, 2$ **do**
    $i \leftarrow j_k$.
    if $\text{Lb}_i < \text{Ub}$ **then**
      ACTIVE $\leftarrow$ ACTIVE $\cup \{N_i\}$.
    end if
  end for
  ACTIVE $\leftarrow$ ACTIVE $\setminus \{N_j\}$.
  if ACTIVE $= \emptyset$ **then**
    break.
  end if
  $\text{Lb} \leftarrow \min_{N_i \in \text{ACTIVE}} \text{Lb}_i$.
end while

4. A simple example

In this section we demonstrate the approach by solving the following toy problem:

$$
\begin{aligned}
\min_{x_1, x_2, x_3} \quad & 3x_1 - 3x_2 + 3x_3 + (x_1 + x_2 + 1)\exp(x_1) + (x_1 + x_2 + 1)\exp(x_3) \\
\text{s.t.} \quad & x_1 + x_2 \geq -1, \\
& x_i \leq 10, \qquad\qquad i \in \{1, 2, 3\}, \\
& \exp(x_2 - x_3) \leq x_1, \\
& 2\exp\left(\tfrac{-x_1}{2}\right) + 2\exp\left(\tfrac{-x_2}{2}\right) \leq 2 + \exp(-1).
\end{aligned} \tag{16}
$$

Let $\mathcal{X}_\mathrm{T}$ denote the feasible set of toy problem (16), consisting of four linear constraints and two convex exponential constraints. The objective is nonconvex, however it is SLC, hence we can apply the proposed framework to find the global optimum.

Linear × Linear. First, we perspectify the SLC objective, and then we consider adding constraints:

$$\begin{aligned}
&(x_1+x_2+1)^2 = x_1^2+x_2^2+2x_1x_2+2x_1+2x_2+1 \geq 0, && \\
&(x_i-10)(x_{i'}-10) = x_ix_{i'}-10x_i-10x_{i'}+100 \geq 0, && i \leq i' \in \{1,2,3\}, \\
&(10-x_i)(x_1+x_2+1) = 10x_1+10x_2+10-x_ix_1-x_ix_2-x_i \geq 0, && i \in \{1,2,3\}, \\
&x_i^2 \geq 0, && i \in \{1,2,3\}.
\end{aligned}$$

Finally, the product of variables x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3 and x_2x_3 in both the perspectified objective as well as the additional generated constraint are substituted by continuous variables X_{11}, X_{22}, X_{33}, X_{12}, X_{13}, and $X_{23} \in \mathbb{R}$ respectively to obtain the following convex relaxation:

$$\begin{array}{lll}
\min\limits_{\boldsymbol{x},\boldsymbol{X}} & 3x_1-3x_2+3x_3+(x_1+x_2+1)\exp\left(\frac{X_{11}+X_{12}+x_1}{x_1+x_2+1}\right)+(x_1+x_2+1)\exp\left(\frac{X_{13}+X_{23}+x_3}{x_1+x_2+1}\right) & \\
\text{s.t.} & \boldsymbol{x} \in \mathcal{X}_\mathrm{T}, & \\
& X_{11}+2X_{12}+X_{22}+2x_1+2x_2+1 \geq 0, & \\
& X_{ii'}-10x_i-10x_{i'}+100 \geq 0, & i \leq i' \in \{1,2,3\}, \\
& 10x_1+10x_2+10-X_{i1}-X_{i2}-x_i \geq 0, & i \in \{1,2,3\}, \\
& X_{ii} \geq 0, & i \in \{1,2,3\}.
\end{array} \tag{17}$$

The solution of (17) appears to be

$$\boldsymbol{x}' = \begin{bmatrix} 1 \\ 1.10 \\ 1.10 \end{bmatrix} \quad \text{and} \quad \boldsymbol{X}' = \begin{bmatrix} X'_{11} & X'_{12} & X'_{13} \\ X'_{21} & X'_{22} & X'_{23} \\ X'_{31} & X'_{32} & X'_{33} \end{bmatrix} = \begin{bmatrix} 12.94 & -48.08 & -41.10 \\ -48.08 & 78.01 & -40.07 \\ -41.10 & -40.07 & 0 \end{bmatrix},$$

with objective value 3, which constitutes a lower bound on the optimal value of (16). Since $\mathcal{X}_\mathrm{T}$ consists of only convex constraints, the obtained $\boldsymbol{x}'$ is contained in the set of feasible candidate solutions to (16), and its corresponding objective value is 20.796, which constitutes an upper bound on the optimal value of (16).

Linear × Convex. Let $\mathcal{X}_\mathrm{TLL}$ denote the feasible set of (17). We pairwise multiply the linear with the nonlinear constraints and obtain the SLC constraints

$$\begin{aligned}
&(x_1+x_2+1)\exp(x_2-x_3) \leq (x_1+x_2+1)x_1, && \\
&(x_1+x_2+1)2\exp\left(\frac{-x_1}{2}\right)+(x_1+x_2+1)2\exp\left(\frac{-x_2}{2}\right) \leq (x_1+x_2+1)(2+\exp(-1)), && \\
&(10-x_i)\exp(x_2-x_3) \leq (10-x_i)x_1, && i \in \{1,2,3\}, \\
&(10-x_i)2\exp\left(\frac{-x_1}{2}\right)+(10-x_i)2\exp\left(\frac{-x_2}{2}\right) \leq (10-x_i)(2+\exp(-1)), && i \in \{1,2,3\}.
\end{aligned}$$

Next, the nonconvex components in the LHS of the above SLC constraints can be reformulated as:

$$
\begin{aligned}
&(x_1+x_2+1)\exp(x_2-x_3) = (x_1+x_2+1)\exp\left(\frac{x_1x_2-x_1x_3+x_2^2-x_2x_3+x_2-x_3}{x_1+x_2+1}\right),\\
&(x_1+x_2+1)2\exp\left(\frac{-x_1}{2}\right) = 2(x_1+x_2+1)\exp\left(\frac{-x_1^2-x_1x_2-x_1}{2(x_1+x_2+1)}\right),\\
&(x_1+x_2+1)2\exp\left(\frac{-x_2}{2}\right) = 2(x_1+x_2+1)\exp\left(\frac{-x_1x_2-x_2^2-x_2}{2(x_1+x_2+1)}\right),\\
&(10-x_i)\exp(x_2-x_3) = (10-x_i)\exp\left(\frac{10x_2-10x_3-x_ix_2+x_ix_3}{10-x_i}\right), && i\in\{1,2,3\},\\
&(10-x_i)2\exp\left(\frac{-x_1}{2}\right) = 2(10-x_i)\exp\left(\frac{-10x_1+x_ix_1}{2(10-x_i)}\right), && i\in\{1,2,3\},\\
&(10-x_i)2\exp\left(\frac{-x_2}{2}\right) = 2(10-x_i)\exp\left(\frac{-10x_2+x_ix_2}{2(10-x_i)}\right), && i\in\{1,2,3\}.
\end{aligned}
$$

Finally, all the product of variables x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3 and x_2x_3 are substituted with newly introduced variables X_{11}, X_{22}, X_{33}, X_{12}, X_{13}, and X_{23} respectively. The convex relaxation that results from the RPT approach is therefore:

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{X}} \quad & 3x_1-3x_2+3x_3+(x_1+x_2+1)\exp\left(\frac{X_{11}+X_{12}+x_1}{x_1+x_2+1}\right)\\
& +\ (x_1+x_2+1)\exp\left(\frac{X_{13}+X_{23}+x_3}{x_1+x_2+1}\right)\\
\text{s.t.} \quad & \boldsymbol{x}\in\mathcal{X}_{\text{TLL}},\\
& (x_1+x_2+1)\exp\left(\frac{X_{12}-X_{13}+X_{22}-X_{23}+x_2-x_3}{x_1+x_2+1}\right)\le X_{11}+X_{12}+x_1,\\
& 2(x_1+x_2+1)\exp\left(\frac{-X_{11}-X_{12}-x_1}{2(x_1+x_2+1)}\right)+2(x_1+x_2+1)\exp\left(\frac{-X_{12}-X_{22}-x_2}{2(x_1+x_2+1)}\right)\\
& \qquad\qquad \le (2+\exp(-1))(x_1+x_2+1),\\
& (10-x_i)\exp\left(\frac{10x_2-10x_3-X_{2i}+X_{3i}}{10-x_i}\right)\le 10x_1-X_{1i}, \qquad i\in\{1,2,3\},\\
& 2(10-x_i)\exp\left(\frac{-10x_1+X_{1i}}{2(10-x_i)}\right)+2(10-x_i)\exp\left(\frac{-10x_2+X_{2i}}{2(10-x_i)}\right)\\
& \qquad\qquad \le (2+\exp(-1))(10-x_i), \qquad i\in\{1,2,3\}.
\end{aligned}
\tag{18}
$$

The solution of (18) is

$$
\boldsymbol{x}' = \begin{bmatrix}1.17\\0.93\\0.77\end{bmatrix} \quad \text{and} \quad \boldsymbol{X}' = \begin{bmatrix}X'_{11} & X'_{12} & X'_{13}\\ X'_{21} & X'_{22} & X'_{23}\\ X'_{31} & X'_{32} & X'_{33}\end{bmatrix} = \begin{bmatrix}1.77 & 0.75 & 0.23\\ 0.75 & 1.16 & 1.31\\ 0.23 & 1.31 & 0.57\end{bmatrix},
$$

with objective value 19.778, which constitutes a tighter lower bound on the optimal value of (16) than (17). The obtained $\boldsymbol{x}'$ is contained in the set of feasible candidate solutions to (16), and its corresponding objective value is 19.809, which constitutes a tighter upper bound on the optimal value of (16) than (17).

Set of candidate solutions. We have the following candidate solutions:

$$\boldsymbol{x}' = \begin{pmatrix} 1.17 \\ 0.93 \\ 0.77 \end{pmatrix}, \ \boldsymbol{x}_1^X = \begin{pmatrix} 1.51 \\ 0.64 \\ 0.20 \end{pmatrix}, \ \boldsymbol{x}_2^X = \begin{pmatrix} 0.81 \\ 1.25 \\ 1.22 \end{pmatrix}, \ \boldsymbol{x}_3^X = \begin{pmatrix} 0.30 \\ 1.47 \\ 0.61 \end{pmatrix}.$$

Observe that only $\boldsymbol{x}'$ is feasible, hence the set of candidate feasible solutions is given by $\mathcal{X}'' = \{\boldsymbol{x}'\}$.

Branch and bound step. We next illustrate the branching mechanism of Section 3.5 on the toy problem (16), with and without enforcing the semidefinite constraint

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{x} \\ \boldsymbol{x}^\top & 1 \end{pmatrix} \succeq \mathbf{0}.$$

In all runs below, we use the same stopping tolerances, namely $\delta = 10^{-4}$ and $\texttt{REL\_GAP\_TOL} = 10^{-4}$, and terminate when either the absolute gap is below δ or the relative gap is below $\texttt{REL\_GAP\_TOL}$.

We first consider the case without the semidefinite constraint. At the root node, we obtain

$$\mathrm{Lb}_0 = 19.778, \qquad \mathrm{Ub}_0 = 19.809.$$

The first branching step corresponds to a positive largest absolute eigenvalue of $\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top$, with

$$\boldsymbol{\nu} = (0.62, -0.54, -0.56)^\top, \qquad l = -0.21,$$

and therefore yields secant splits:

$$\boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu} \leq -10.65\,\boldsymbol{\nu}^\top \boldsymbol{x} - 2.16 \quad (N_1), \qquad \boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu} \leq 7.68\,\boldsymbol{\nu}^\top \boldsymbol{x} + 1.63 \quad (N_2).$$

Solving the two child relaxations yields

$$(\mathrm{Lb}_1, \mathrm{Ub}_1) = (19.790, 19.790), \qquad (\mathrm{Lb}_2, \mathrm{Ub}_2) = (19.787, 19.788).$$

Hence, the global bounds are

$$\mathrm{LB} = 19.787, \qquad \mathrm{UB} = 19.788.$$

With $\texttt{REL\_GAP\_TOL} = 10^{-4}$, the relative-gap criterion is satisfied after one branching iteration, and the returned incumbent feasible value is 19.788 with

$$\hat{\boldsymbol{x}} \approx (1.19, 0.92, 0.75)^\top.$$

When the semidefinite constraint is enforced, the root-node RPT relaxation and upper-bounding procedure yield

$$\mathrm{Lb}_0 = 19.784, \qquad \mathrm{Ub}_0 = 19.792.$$

The first branching step uses

$$\boldsymbol{\nu} = (-0.46,\, 0.40,\, 0.79)^\top, \qquad l = 0.43,$$

and the largest absolute eigenvalue is positive. Hence, Algorithm 1 again generates secant splits (rather than SOC splits):

$$\boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu} \le -6.39\,\boldsymbol{\nu}^\top \boldsymbol{x} + 2.91 \quad (N_1), \qquad \boldsymbol{\nu}^\top \boldsymbol{X}\boldsymbol{\nu} \le 11.91\,\boldsymbol{\nu}^\top \boldsymbol{x} - 4.90 \quad (N_2).$$

Solving the two child relaxations yields

$$(\mathrm{Lb}_1, \mathrm{Ub}_1) = (19.787,\, 19.787), \qquad (\mathrm{Lb}_2, \mathrm{Ub}_2) = (19.787,\, 19.787).$$

Thus, the global bounds satisfy

$$\mathrm{LB} = 19.787, \qquad \mathrm{UB} = 19.787,$$

and the algorithm again terminates after one branching iteration. In both runs, the selected split is secant, and no SOC split is activated.

Remark 3. The behavior observed in this toy example, that is, a small search tree and several nodes where the relaxation closes the gap (here, LB = UB), is qualitatively consistent with the computational behavior reported for eigenvector branching in Anstreicher (2022), where many instances are solved with shallow trees and very small final optimality gaps.

5. Improving the RPT-BB approach

In this section, we describe several ways to strengthen the RPT-BB approach.

5.1. Multiplying with redundant linear constraints

We show that adding linear constraints that are redundant to existing linear constraints does not tighten the RPT relaxation, while adding linear constraints that are redundant to existing nonlinear constraints might be useful.

Definition 2 (Redundant inequality constraints). An inequality constraint $f(\boldsymbol{x}) \le 0$, where $f: \mathbb{R}^{n_x} \to (-\infty, +\infty]$, is *redundant* to a nonempty set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, if $\mathcal{X} \subseteq \{\boldsymbol{x} \mid f(\boldsymbol{x}) \le 0\}$.

Theorem 2. *Let the linear constraint $\boldsymbol{d}^\top \boldsymbol{x} \le q$, where $\boldsymbol{d} \in \mathbb{R}^{n_x}$ with $\boldsymbol{d} \neq \boldsymbol{0}$ and $q \in \mathbb{R}$, be redundant to $\{\boldsymbol{x} \mid \boldsymbol{B}^\top \boldsymbol{x} \le \boldsymbol{p}\} \neq \emptyset$, where $\boldsymbol{B} \in \mathbb{R}^{n_x \times L}$ and $\boldsymbol{p} \in \mathbb{R}^L$. Then, the constraints $\boldsymbol{d}^\top \boldsymbol{x} \le q$, $2q\boldsymbol{d}^\top \boldsymbol{x} \le \boldsymbol{d}^\top \boldsymbol{X}\boldsymbol{d} + q^2$, and $(q - \boldsymbol{d}^\top \boldsymbol{x}) f\left(\frac{q\boldsymbol{x} - \boldsymbol{X}\boldsymbol{d}}{q - \boldsymbol{d}^\top \boldsymbol{x}}\right) \le 0$, are redundant to*

$$\left\{ (\boldsymbol{x}, \boldsymbol{X}) \;\middle|\; \begin{array}{l} \boldsymbol{B}^\top \boldsymbol{x} \le \boldsymbol{p} \\ \boldsymbol{p}\boldsymbol{x}^\top \boldsymbol{B} + \boldsymbol{B}^\top \boldsymbol{x}\boldsymbol{p}^\top \le \boldsymbol{B}^\top \boldsymbol{X}\boldsymbol{B} + \boldsymbol{p}\boldsymbol{p}^\top \\ (p_\ell - \boldsymbol{b}_\ell^\top \boldsymbol{x}) f\left(\frac{p_\ell \boldsymbol{x} - \boldsymbol{X}\boldsymbol{b}_\ell}{p_\ell - \boldsymbol{b}_\ell^\top \boldsymbol{x}}\right) \le 0, \;\; \ell \in \mathcal{L} \end{array} \right\}.$$

Here, $f:\mathbb{R}^{n_x}\to(-\infty,+\infty]$ is proper, closed and convex; $\boldsymbol{b}_\ell$ is the ℓ-th column of the matrix $\boldsymbol{B}$; and the last two inequalities result from pairwise multiplication of the linear constraints $p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x}$ with itself and $f(\boldsymbol{x})\le 0$, respectively.

Proof. Assume that $\boldsymbol{d}^\top\boldsymbol{x}\le q$ is redundant to $\{\boldsymbol{x}\mid\boldsymbol{B}^\top\boldsymbol{x}\le\boldsymbol{p}\}$, then the optimal values of

$$\begin{array}{ll}\min\limits_{\boldsymbol{x}} & q-\boldsymbol{d}^\top\boldsymbol{x}\\ \text{s.t.} & \boldsymbol{B}^\top\boldsymbol{x}\le\boldsymbol{p}\end{array}\qquad\text{and}\qquad\begin{array}{ll}\max\limits_{\boldsymbol{y}\ge\boldsymbol{0}} & q-\boldsymbol{p}^\top\boldsymbol{y}\\ \text{s.t.} & \boldsymbol{B}\boldsymbol{y}=\boldsymbol{d}\end{array}$$

coincide and both are nonnegative thanks to the strong duality of linear optimization and the redundancy of $\boldsymbol{d}^\top\boldsymbol{x}\le q$ to $\{\boldsymbol{x}\mid\boldsymbol{B}^\top\boldsymbol{x}\le\boldsymbol{p}\}$, which implies that there exists a $\boldsymbol{y}\in\mathbb{R}^L_+$ such that $\boldsymbol{d}^\top\boldsymbol{x}\le q$ is redundant to $\{\boldsymbol{x}\mid\boldsymbol{b}_y^\top\boldsymbol{x}\le p_y\}$, where $\boldsymbol{b}_y=\boldsymbol{B}\boldsymbol{y}=\boldsymbol{d}$ and $p_y=\boldsymbol{p}^\top\boldsymbol{y}\le q$. Then, for any $\boldsymbol{x}$ that satisfies $\boldsymbol{b}_y^\top\boldsymbol{x}\le p_y$ and $f(\boldsymbol{x})\le 0$, we have that

$$(q-\boldsymbol{d}^\top\boldsymbol{x})f\left(\frac{(q-\boldsymbol{d}^\top\boldsymbol{x})\boldsymbol{x}}{q-\boldsymbol{d}^\top\boldsymbol{x}}\right)\le 0\quad\text{and}\quad(p_y-\boldsymbol{b}_y^\top\boldsymbol{x})f\left(\frac{(p_y-\boldsymbol{b}_y^\top\boldsymbol{x})\boldsymbol{x}}{p_y-\boldsymbol{b}_y^\top\boldsymbol{x}}\right)\le 0.$$

Moreover, for any $\boldsymbol{X}\in\mathbb{S}^{n_x}$ we have

$$(q-\boldsymbol{d}^\top\boldsymbol{x})f\left(\frac{q\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}}{q-\boldsymbol{d}^\top\boldsymbol{x}}\right)\le 0\quad\Longleftrightarrow\quad(p_y-\boldsymbol{b}_y^\top\boldsymbol{x})f\left(\frac{p_y\boldsymbol{x}-\boldsymbol{X}\boldsymbol{b}_y}{p_y-\boldsymbol{b}_y^\top\boldsymbol{x}}\right)\le 0,$$

because $\boldsymbol{b}_y=\boldsymbol{d}$ so that $\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{d}=\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{b}_y$ and $\boldsymbol{X}\boldsymbol{d}=\boldsymbol{X}\boldsymbol{b}_y$. Notice that

$$\begin{aligned}(p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x})f\left(\frac{p_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{b}_\ell}{p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x}}\right)\le 0,\ \ell\in\mathcal{L}\quad&\Longrightarrow\quad\sum_{\ell\in\mathcal{L}}\theta_\ell(p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x})f\left(\frac{p_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{b}_\ell}{p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x}}\right)\le 0\\ &\Longrightarrow\quad\left(\sum_{\ell\in\mathcal{L}}\theta_\ell(p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x})\right)f\left(\frac{\sum_{\ell\in\mathcal{L}}\theta_\ell(p_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{b}_\ell)}{\sum_{\ell\in\mathcal{L}}\theta_\ell(p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x})}\right)\le 0\\ &\Longrightarrow\quad(p_y-\boldsymbol{b}_y^\top\boldsymbol{x})f\left(\frac{p_y\boldsymbol{x}-\boldsymbol{X}\boldsymbol{b}_y}{p_y-\boldsymbol{b}_y^\top\boldsymbol{x}}\right)\le 0,\end{aligned}$$

where $\theta_\ell=y_\ell/\sum_{\ell\in\mathcal{L}}y_\ell$ for all $\ell\in\mathcal{L}$ (note that $\boldsymbol{\theta}\in\mathbb{R}^L_+$ and $\sum_{\ell\in\mathcal{L}}\theta_\ell=1$). Here, the second implication follows from the convexity of the perspective functions. Therefore, the constraint $(q-\boldsymbol{d}^\top\boldsymbol{x})f\left(\frac{q\boldsymbol{x}-\boldsymbol{X}\boldsymbol{d}}{q-\boldsymbol{d}^\top\boldsymbol{x}}\right)\le 0$ is redundant to

$$\left\{\boldsymbol{x}\;\middle|\;\boldsymbol{b}_y^\top\boldsymbol{x}\le p_y,\ (p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x})f\left(\frac{p_\ell\boldsymbol{x}-\boldsymbol{X}\boldsymbol{b}_\ell}{p_\ell-\boldsymbol{b}_\ell^\top\boldsymbol{x}}\right)\le 0,\ \ell\in\mathcal{L}\right\}.$$

Thus, the claim follows. □

REMARK 4. Theorem 2 parallels a standard observation in the RLT literature: adding a linear inequality that is redundant for the original linear description does not strengthen the RLT relaxation; see, e.g., Proposition 1(i) in Hof and Walter (2025).

Note that adding linear constraints that are redundant to existing nonlinear constraints might be useful, as is demonstrated in Example 6.

Example 6. Consider the following nonconvex problem:

$$\begin{array}{ll} \min_{x_1,x_2} & 2x_1+3x_2-5x_1x_2-(x_1+2)\ln(x_1+2) \\ \text{s.t.} & x_1+x_2\leq 1, \\ & \exp(-x_1)+\exp(-x_2)\leq 1+\exp(-1). \end{array} \tag{19}$$

Note that $\ln(x_1+2)$ is well defined if $x_1>-2$, which is ensured by the second inequality. The objective contains a sum of two SLC functions, namely, $-5x_1x_2$ and $-(x_1+2)\ln(x_1+2)$. The obtained convex relaxation of Problem (19) from RPT without the optional SDP relaxation has an objective value of -35.17. The obtained optimal solution $\boldsymbol{x}'=(1,0)^\top$ is a feasible solution to (19), and its corresponding objective value is 1.30, which constitutes an upper bound on the optimal value of (19).

The linear constraints $x_1\geq -1$ and $x_2\geq -1$ are redundant to the second inequality. However, adding those constraints to (19) and subsequently applying RPT results in a convex relaxation of Problem (19) with objective value of -4.47. Again, the obtained optimal solution $\boldsymbol{x}'=(0.5,0.5)^\top$ is a feasible solution to (19), and its corresponding objective value is -1.04, which constitutes an upper bound on the optimal value of (19). Hence, by adding the redundant linear constraints $x_1\geq -1$ and $x_2\geq -1$, we obtain a tighter lower- and upper bound on the optimal objective value of (19). □

Remark 5. Example 6 illustrates the strengthening effect discussed in Section 3.2: even linear inequalities that are redundant for the original feasible set can tighten the lifted RPT relaxation, because they generate additional valid product constraints when combined with other (nonlinear) constraints. Related effects are exploited in the SOC–RLT literature for QCQP-type models; see, e.g., Burer and Anstreicher (2013).

5.2. Handling biconvex problems with convex constraints

We consider the following generic instance of (1):

$$\begin{array}{ll} \min_{\boldsymbol{x},\boldsymbol{y}} & c_0(\boldsymbol{x})+\sum_k r_k(\boldsymbol{y})c_k(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\mathcal{Y}, \end{array} \tag{20}$$

where the functions $r_k(\cdot)$ and $c_k(\cdot)$ are assumed to be convex, and $r_k(\boldsymbol{y}),c_k(\boldsymbol{x})\geq 0$. We show that the constraints in the RPT relaxation of (20) resulting from pairwise multiplication in the same set, i.e., either in $\mathcal{X}$ or in $\mathcal{Y}$, are redundant as long as the LMI as given in Section 3.3 is not included.

Lemma 4. *The additional constraints in the RPT relaxation of (20) resulting from pairwise multiplication in $\mathcal{X} \times \mathcal{X}$ and $\mathcal{Y} \times \mathcal{Y}$ are redundant, if the additional constraints resulting from pairwise multiplication in $\mathcal{X} \times \mathcal{Y}$ are included, and the following LMI*

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{U} & \boldsymbol{x} \\ \boldsymbol{U}^\top & \boldsymbol{Y} & \boldsymbol{y} \\ \boldsymbol{x}^\top & \boldsymbol{y}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0},$$

where $\boldsymbol{x}\boldsymbol{x}^\top$, $\boldsymbol{x}\boldsymbol{y}^\top$ and $\boldsymbol{y}\boldsymbol{y}^\top$ are linearized by $\boldsymbol{X}$, $\boldsymbol{U} \in \mathbb{R}^{n_x \times n_y}$ and $\boldsymbol{Y} \in \mathbb{R}^{n_y \times n_y}$, respectively, is not included in the RPT relaxation.

Proof. Notice that any feasible solution for the problem involving all constraint multiplications is also feasible for the one involving only those in $\mathcal{X} \times \mathcal{Y}$. On the other hand, if a solution is feasible for the problem involving only the multiplications in $\mathcal{X} \times \mathcal{Y}$, since we are not using the LMI, we can take $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$, $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{y}^\top$ and therefore have a feasible solution for the problem involving all multiplications. Therefore, we conclude that the two formulations are equivalent, which shows that the constraint multiplications in $\mathcal{X} \times \mathcal{X}$ and $\mathcal{Y} \times \mathcal{Y}$ are redundant. □

Observe that the problem of minimizing a DC function is a special case of (20) where $\mathcal{Y} = \operatorname{dom}(c_1^*)$ and both r_k and c_k are linear, as a result of the biconjugate reformulation as explained in Example 1. Finally, we have the following remark in terms of branching, when we solve Problem (20) with Algorithm 2.

Remark 6. When we apply Algorithm 2 to minimization problems of the form (20) in which we include the LMI, we typically generate hyperplanes by computing the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top$, i.e., we only generate hyperplanes in the $\boldsymbol{x}$-space. Alternatively, hyperplanes can be generated in the $\boldsymbol{x}\boldsymbol{y}$-space by considering the eigenvectors of the matrix

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{U} \\ \boldsymbol{U}^\top & \boldsymbol{Y} \end{pmatrix} - (\boldsymbol{x}, \boldsymbol{y})\,(\boldsymbol{x}, \boldsymbol{y})^\top,$$

where $(\boldsymbol{x}, \boldsymbol{y})$ denotes the vector formed by stacking $\boldsymbol{x}$ and $\boldsymbol{y}$. We observe that also generating hyperplanes in the $\boldsymbol{y}$-space adds many more constraints in each branch and bound iteration, which can increase the computation time at each successive child node. On the other hand, generating hyperplanes in the $\boldsymbol{y}$-space might reduce the total number of hyperplanes that need to be generated, potentially decreasing the overall computation time. The question of whether generating hyperplanes in the $\boldsymbol{y}$-space could be beneficial is left for future research. Finally, we note that when we apply Algorithm 2 to minimization problems of the form (20) in which we do not include the LMI, one can branch based on left and right directions associated with the matrix $\boldsymbol{R} = \boldsymbol{U} - \boldsymbol{x}\boldsymbol{y}^\top$. Concretely, let $\boldsymbol{v}^r$ be a unit eigenvector corresponding to the largest eigenvalue of $\boldsymbol{R}^\top \boldsymbol{R}$, i.e.,

$$\boldsymbol{R}^\top \boldsymbol{R} \boldsymbol{v}^r = \lambda_{\max}(\boldsymbol{R}^\top \boldsymbol{R})\, \boldsymbol{v}^r.$$

If $\boldsymbol{R}\boldsymbol{v}^r \neq \boldsymbol{0}$, define $\boldsymbol{v}^\ell = \boldsymbol{R}\boldsymbol{v}^r/\|\boldsymbol{R}\boldsymbol{v}^r\|$; then $\boldsymbol{v}^\ell$ is a unit eigenvector corresponding to the largest eigenvalue of $\boldsymbol{R}\boldsymbol{R}^\top$, i.e.,

$$\boldsymbol{R}\boldsymbol{R}^\top \boldsymbol{v}^\ell = \lambda_{\max}(\boldsymbol{R}\boldsymbol{R}^\top)\,\boldsymbol{v}^\ell.$$

□

5.3. Strengthening upper bounds with eigenvectors of the optimal solution

At optimality we will always have $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$. If we multiply both sides with $\boldsymbol{x}$ we obtain $\boldsymbol{X}\boldsymbol{x} = (\boldsymbol{x}^\top\boldsymbol{x})\boldsymbol{x}$. Therefore, we notice that $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{X}$ with corresponding eigenvalue $\boldsymbol{x}^\top\boldsymbol{x}$. Hence, we can add the eigenvectors of $\boldsymbol{X}$ to the set of candidate feasible solutions $\mathcal{X}'$, as described in Section 3.4. Example 7 illustrates a case where the tightest upper bound can be obtained from an eigenvector of $\boldsymbol{X}$.

Example 7. Consider the following toy problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & (x_1 + x_2 + 1)\exp(x_2) \\ \text{s.t.} & x_1 + x_2 + 1 \geq 0, \\ & x_1 x_2 \geq -1. \end{array} \tag{21}$$

The optimal solution of (21) is $(-1, 0)$ with optimal value 0. After applying RPT we obtain the following relaxation

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & (x_1 + x_2 + 1)\exp\left(\frac{X_{12}+X_{22}+x_2}{x_1+x_2+1}\right) \\ \text{s.t.} & x_1 + x_2 + 1 \geq 0, \\ & X_{12} \geq -1, \\ & X_{11} + X_{22} + 2X_{12} + 2x_1 + 2x_2 + 1 \geq 0. \end{array} \tag{22}$$

The optimal solution of (22) is

$$\boldsymbol{x}^* = \begin{bmatrix} 0.71 \\ -1.57 \end{bmatrix} \quad \text{and} \quad \boldsymbol{X}^* = \begin{bmatrix} 2.73 & -1.00 \\ -1.00 & 0.00 \end{bmatrix},$$

with optimal value 0, which gives us a lower bound for the optimal value of (21). We have the following candidate vectors

$$\boldsymbol{x}_1^X = \frac{\boldsymbol{X}_1^*}{x_1^*} = \begin{bmatrix} 3.85 \\ -1.41 \end{bmatrix}, \quad \boldsymbol{x}_2^X = \frac{\boldsymbol{X}_2^*}{x_2^*} = \begin{bmatrix} 0.64 \\ 0.00 \end{bmatrix}.$$

The eigenvectors of $\boldsymbol{X}^*$ are

$$\boldsymbol{x}_3^{EV} = \begin{bmatrix} -0.31 \\ -0.95 \end{bmatrix}, \quad \boldsymbol{x}_4^{EV} = \begin{bmatrix} -0.95 \\ 0.31 \end{bmatrix}.$$

We observe that $\boldsymbol{x}^*$ is infeasible as $x_1^* x_2^* = -1.11 < -1$. Moreover, $\boldsymbol{x}_1^X$ is infeasible as $(x_1^X)_1 (x_1^X)_2 = -5.44 < -1$ and $\boldsymbol{x}_3^{EV}$ is infeasible as $(x_3^{EV})_1 + (x_3^{EV})_2 + 1 = -0.26 < 0$. Finally, we notice that $\boldsymbol{x}_2^X$ is feasible and gives an upper bound of 1.64, while $\boldsymbol{x}_4^{EV}$ is also feasible and gives an upper bound of 0.49. Therefore, in this example the tightest upper bound is obtained from the second eigenvector of $\boldsymbol{X}^*$. □

6. Convergence analysis of the RPT-BB approach

In deterministic global optimization, convergence of branch and bound (B&B) algorithms that rely on dual (lower) bounds is typically established under three structural ingredients: (i) a valid lower-bounding procedure based on relaxations, (ii) an upper-bounding mechanism that maintains the best known feasible objective value, and (iii) an exhaustive refinement rule, meaning that along any infinite sequence of nested nodes, the associated regions are refined so that their diameters tend to zero. This type of requirement is standard; see, e.g., Dür (2001) and the references therein. In the RLT-based global-optimization literature, convergence proofs are commonly organized around the same template, with the main technical burden being to show that the node relaxations are asymptotically exact under refinement, i.e., along any nested sequence of nodes whose regions shrink to a point, the relaxation gap vanishes (equivalently, the relaxation optimal values converge to the objective value at the limiting point); see, e.g., Sherali and Alameddine (1992); Sherali and Tuncbilek (1992).

The practical branching scheme in Section 3.5 is designed to highlight the effect of eigenvector branching and is the one used in the numerical experiments. For the purpose of a general convergence proof, however, it is convenient to consider a variant that enforces an explicit exhaustive spatial refinement step. We therefore introduce below a periodic coordinate-refinement (Adaptation A) and establish convergence for Algorithm 2 with this adaptation. This adaptation is included only for the theoretical analysis; it was not incorporated into the computational study, so as not to confound the performance assessment of the proposed eigenvector branching mechanism.

Adaptation A (periodic exhaustive coordinate refinement). Fix $d \in \mathbb{Z}_{++}$, and define $\bar{\mathcal{J}} := \{kd : k \in \mathbb{Z}_{++}\}$. For every depth $j \in \bar{\mathcal{J}}$ and every leaf node N at depth j, we replace the eigenvector branching rule by a longest-edge coordinate split in $\boldsymbol{x}$-space, constructed as follows.

For a node N with region $\mathcal{X}_N \subseteq \mathbb{R}^{n_x} \times \mathbb{S}^{n_x}$, define its $\boldsymbol{x}$-projection by

$$\Pi_x(\mathcal{X}_N) := \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \exists \boldsymbol{X} \in \mathbb{S}^{n_x} : \ (\boldsymbol{x}, \boldsymbol{X}) \in \mathcal{X}_N\}.$$

By Assumption 1, $\Pi_x(\mathcal{X}_N)$ is bounded. We compute valid coordinate bounds

$$x_i^{\min}(N) := \min\{x_i : \boldsymbol{x} \in \Pi_x(\mathcal{X}_N)\}, \qquad x_i^{\max}(N) := \max\{x_i : \boldsymbol{x} \in \Pi_x(\mathcal{X}_N)\}, \qquad i = 1, \ldots, n_x,$$

and choose

$$i' \in \arg\max_{i=1,\ldots,n_x} \big(x_i^{\max}(N) - x_i^{\min}(N)\big).$$

We then branch on

$$x_{i'} = \frac{1}{2}\big(x_{i'}^{\max}(N) + x_{i'}^{\min}(N)\big),$$

i.e., we create the two children by adding $x_{i'} \leq \frac{1}{2}(x_{i'}^{\max}(N) + x_{i'}^{\min}(N))$ and $x_{i'} \geq \frac{1}{2}(x_{i'}^{\max}(N) + x_{i'}^{\min}(N))$, respectively.

THEOREM 3. *If for each $i \in \mathcal{I}$ and $k \in \mathcal{K}_0$, the function c_{ik} and its corresponding recession function $\delta^*_{\mathrm{dom}(c^*_{ik})}$ are Lipschitz continuous, then Algorithm 2 with Adaptation A converges to a global optimal solution of Problem (1).*

Proof. We verify the standard ingredients for convergence of a dual-bounding spatial B&B method.

First, at every node, the RPT formulation solved by Algorithm 2 is a valid convex relaxation of Problem (1) with the node restrictions included, hence the node optimal value is a valid lower bound. Second, the upper-bounding procedure in Section 3.4 produces feasible solutions of Problem (1), hence valid upper bounds. It remains to prove asymptotic exactness of the node relaxations along any infinite nested branch.

Let $\{N_r\}_{r\geq 1}$ be any infinite nested branch of the B&B tree. Because Adaptation A is applied at all depths in $\bar{\mathcal{J}}$, there exists an infinite subsequence $\{N_{r_t}\}_{t\geq 1}$ of nodes on this branch at depths in $\bar{\mathcal{J}}$, and each N_{r_t} is branched by the longest-edge coordinate rule of Adaptation A. For simplicity, define

$$\alpha_i^t := \frac{1}{2}\big(x_i^{\max}(N_{r_t}) + x_i^{\min}(N_{r_t})\big), \qquad w_i^t := x_i^{\max}(N_{r_t}) - x_i^{\min}(N_{r_t}), \qquad \epsilon_t := \frac{1}{2}\max_{i=1,\dots,n_x} w_i^t.$$

Then, by construction,

$$\Pi_x(\mathcal{X}_{N_{r_t}}) \subseteq \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \;\middle|\; \|\boldsymbol{x} - \boldsymbol{\alpha}^t\|_\infty \leq \epsilon_t \right\}, \tag{23}$$

where $\boldsymbol{\alpha}^t := (\alpha_1^t, \dots, \alpha_{n_x}^t)$.

Since Adaptation A performs longest-edge bisection at each node N_{r_t}, and the branch is nested, the corresponding sequence of enclosing boxes is nested and undergoes infinitely many longest-edge bisections. By the standard exhaustiveness property of longest-edge bisection in finite dimension (see, e.g., Dür (2001)), we have

$$\epsilon_t \to 0 \qquad (t \to \infty). \tag{24}$$

We next show that $\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top$ vanishes asymptotically on the node relaxations along this subsequence. Fix t, and let $(\boldsymbol{x}, \boldsymbol{X})$ be feasible for the node relaxation at N_{r_t}. From (23), for any $p, q \in \{1, \dots, n_x\}$, we have the valid inequalities

$$x_p - \alpha_p^t \leq \epsilon_t, \tag{25}$$

$$x_p - \alpha_p^t \geq -\epsilon_t, \tag{26}$$

$$x_q - \alpha_q^t \leq \epsilon_t, \tag{27}$$

$$x_q - \alpha_q^t \geq -\epsilon_t. \tag{28}$$

These inequalities are redundant for $\Pi_x(\mathcal{X}_{N_{r_t}})$. Hence, by Theorem 2, the inequalities obtained from multiplying them and applying the same RPT/perspectification step are also redundant for the node relaxation.

Multiplying (26) and (28), and applying RPT, yields

$$0 \le (x_p - \alpha_p^t + \epsilon_t)(x_q - \alpha_q^t + \epsilon_t) = X_{pq} - \alpha_p^t x_q + \epsilon_t x_q + (x_p - \alpha_p^t + \epsilon_t)(-\alpha_q^t + \epsilon_t).$$

Rearranging and using (23) (hence $|x_p - \alpha_p^t| \le \epsilon_t$ and $|x_q - \alpha_q^t| \le \epsilon_t$) gives

$$X_{pq} - x_p x_q \ge -4\epsilon_t\big(|\alpha_q^t| + \epsilon_t\big).$$

Similarly, multiplying (26) and (27) and applying RPT yields

$$X_{pq} - x_p x_q \le 4\epsilon_t\big(|\alpha_q^t| + \epsilon_t\big).$$

Therefore,

$$|X_{pq} - x_p x_q| \le 4\epsilon_t\big(|\alpha_q^t| + \epsilon_t\big), \qquad p,q = 1,\dots,n_x. \tag{29}$$

By Assumption 1, there exists $M_X < \infty$ such that $\|\boldsymbol{x}\|_\infty \le M_X$ for all $\boldsymbol{x} \in \mathcal{X}$. Since $x_i^{\min}(N_{r_t})$ and $x_i^{\max}(N_{r_t})$ are bounds over subsets of $\mathcal{X}$, we also have $|\alpha_q^t| \le M_X$ for all q,t. Using (29), we obtain

$$\|\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top\|_F \le n_x \max_{p,q} |X_{pq} - x_p x_q| \le 4 n_x\, \epsilon_t (M_X + \epsilon_t). \tag{30}$$

Combining (30) with (24) shows that

$$\|\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top\|_F \to 0 \qquad (t \to \infty)$$

along the subsequence $\{N_{r_t}\}$.

We now show asymptotic exactness of the perspectified terms. Fix $k \in \mathcal{K}_0$, and let $\tilde{f}_k(\boldsymbol{x}, \boldsymbol{X})$ denote the corresponding perspectified contribution in the RPT relaxation. For $\boldsymbol{X}', \boldsymbol{X}'' \in \mathbb{S}^{n_x}$, and for the case $q_{ik} - \boldsymbol{d}_{ik}^\top \boldsymbol{x} > 0$, Lipschitz continuity of c_{ik} implies

$$\begin{aligned}\big|\tilde{f}_k(\boldsymbol{x}, \boldsymbol{X}') - \tilde{f}_k(\boldsymbol{x}, \boldsymbol{X}'')\big| &\le \sum_i L_{ik}\, \|(\boldsymbol{X}' - \boldsymbol{X}'')\boldsymbol{d}_{ik}\| \\ &\le \left(\sum_i L_{ik} \|\boldsymbol{d}_{ik}\|\right) \|\boldsymbol{X}' - \boldsymbol{X}''\|_F \\ &=: \tilde{L}_k\, \|\boldsymbol{X}' - \boldsymbol{X}''\|_F.\end{aligned}$$

Taking $\boldsymbol{X}' = \boldsymbol{X}$ and $\boldsymbol{X}'' = \boldsymbol{x}\boldsymbol{x}^\top$, and using (30), we obtain

$$\big|\tilde{f}_k(\boldsymbol{x}, \boldsymbol{X}) - \tilde{f}_k(\boldsymbol{x}, \boldsymbol{x}\boldsymbol{x}^\top)\big| \le \tilde{L}_k\, \|\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top\|_F \to 0 \qquad (t \to \infty).$$

When $q_{ik} - \boldsymbol{d}_{ik}^\top \boldsymbol{x} = 0$, the same conclusion follows analogously from the assumed Lipschitz continuity of the recession function $\delta^*_{\mathrm{dom}(c^*_{ik})}$.

Hence, along any infinite nested branch, the node relaxations are asymptotically exact under Adaptation A. Together with valid lower bounds, valid upper bounds, pruning, and best-bound node selection in Algorithm 2, the standard convergence theory for dual-bounding spatial B&B applies and yields convergence to a global optimum of Problem (1). □

Remark 7. Adaptation A is a theoretical safeguard introduced to enforce exhaustive refinement and thereby establish a general convergence guarantee. It was not included in the numerical experiments, whose purpose is to isolate and assess the computational effect of the proposed eigenvector branching strategy.

Theorem 3 requires Lipschitz continuity of the recession functions of c_{ik}. The following example shows that convergence may fail without this assumption:

$$c(x) = xe^x, \qquad x \geq 0.$$

Its perspective function is

$$\tilde{c}(x, x') = xe^{x'/x} \qquad (x > 0),$$

with the usual closed extension at $x = 0$. Now take $x = 0$. Even if $x' \geq 0$ is arbitrarily close to 0, one has $\tilde{c}(0, x') = \infty$ for every $x' > 0$. Thus, continuity (and hence the asymptotic-exactness argument above) can fail at the boundary.

7. Known convex reformulations and relaxations obtained via RPT

In this section, we show that several convex reformulations and relaxations for several classes of nonconvex problems derived in the literature can also be obtained via RPT.

7.1. Disjunctive optimization

A linear description of the convex hull of the union of convex sets can be derived by using RPT. It follows from the definition that the convex hull of the union of nonempty, compact convex sets $\mathcal{X}_k = \{\boldsymbol{x} \mid \boldsymbol{h}_k(\boldsymbol{x}) \leq \boldsymbol{0}\}$, $k \in \mathcal{K}$ is:

$$\mathrm{conv}\left(\bigcup_{k\in\mathcal{K}} \mathcal{X}_k\right) = \left\{\boldsymbol{x} \;\middle|\; \exists \boldsymbol{x}_k \in \mathcal{X}_k, \boldsymbol{\lambda} \geq \boldsymbol{0} : \boldsymbol{x} = \sum_{k\in\mathcal{K}} \lambda_k \boldsymbol{x}_k,\; \sum_{k\in\mathcal{K}} \lambda_k = 1\right\},$$

where $\boldsymbol{h}_k(\boldsymbol{x}) = [h_{1k}(\boldsymbol{x})\; h_{2k}(\boldsymbol{x})\; \cdots\; h_{Jk}(\boldsymbol{x})]^\top$, and $h_{jk} : \mathbb{R}^{n_x} \to (-\infty, +\infty]$ is proper, closed and convex for every $j \in \mathcal{J}$, $k \in \mathcal{K}$. This description is nonlinear and nonconvex, since it contains products of variables $\lambda_k \boldsymbol{x}_k$, $k \in \mathcal{K}$. One can apply RPT to obtain the following convex relaxation

$$\left\{\boldsymbol{x} \;\middle|\; \exists \boldsymbol{u}_k : \boldsymbol{x} = \sum_{k\in\mathcal{K}} \boldsymbol{u}_k,\; \sum_{k\in\mathcal{K}} \lambda_k = 1,\; \boldsymbol{\lambda} \geq \boldsymbol{0},\; \lambda_k \boldsymbol{h}_k(\boldsymbol{u}_k/\lambda_k) \leq \boldsymbol{0},\; k \in \mathcal{K}\right\}.$$

This convex relaxation is exact according to Gorissen et al. (2014, Lemma 1), which applies because $\mathcal{X}_k$, $k \in \mathcal{K}$, are nonempty, compact and convex sets. We now use this observation to derive convex relaxation for disjunctive optimization problems with general convex sets. In Sherali and Adams (2005, Section 4), the authors derive similar result for disjunctive optimization problems with a linear objective function and polyhedral sets $\mathcal{X}_k$, $k \in \mathcal{K}$. Consider a generic disjunctive optimization problem

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in \bigcup_{k \in \mathcal{K}} \mathcal{X}_k, \end{aligned} \tag{DP}$$

where $f : \mathbb{R}^{n_x} \to (-\infty, +\infty]$ is proper, closed and convex. Disjunctive optimization problems are in general nonconvex because its feasible region constitutes a union of convex sets $\mathcal{X}_k$. By applying RPT to the feasible region of (DP), we obtain the following convex relaxation:

$$\begin{aligned} \min_{\boldsymbol{y}, \{\boldsymbol{u}_k\}_k} \quad & f\left(\sum_{k \in \mathcal{K}} \boldsymbol{u}_k\right) \\ \text{s.t.} \quad & y_k \boldsymbol{h}_k(\boldsymbol{u}_k / y_k) \leq \boldsymbol{0}, \qquad k \in \mathcal{K}, \\ & \sum_{k \in \mathcal{K}} y_k = 1, \\ & y_k \geq 0, \qquad k \in \mathcal{K}, \end{aligned}$$

which is often referred to as the hull relaxation (Grossmann and Lee, 2003). Note that this hull relaxation is tight if $f(\cdot)$ is a linear function, and $\mathcal{X}_k$, $k \in \mathcal{K}$, are nonempty, compact and convex sets.

7.2. Generalized linear optimization

Consider a generalized linear optimization problem of the following form (Dantzig, 1963, p. 434):

$$\begin{aligned} \min_{\boldsymbol{y}, \{\boldsymbol{x}_k\}_k} \quad & \boldsymbol{c}^\top \boldsymbol{y} \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}_0} \boldsymbol{x}_k y_k \leq \boldsymbol{b}, \\ & \boldsymbol{y} \geq \boldsymbol{0}, \\ & \boldsymbol{x}_k \in \mathcal{X}_k, \qquad k \in \mathcal{K}_0, \end{aligned} \tag{GLP}$$

where $\mathcal{X}_k = \{\boldsymbol{x} \mid \boldsymbol{h}_k(\boldsymbol{x}) \leq \boldsymbol{0}\}$, $k \in \mathcal{K}_0$, and $\boldsymbol{h}_k : \mathbb{R}^{n_x} \to (-\infty, +\infty]^J$ is a vector of J proper, closed and convex functions for each $k \in \mathcal{K}_0$. The partial RPT relaxation of (GLP) is:

$$\begin{aligned} \min_{\boldsymbol{y}, \{\boldsymbol{u}_k\}_k} \quad & \boldsymbol{c}^\top \boldsymbol{y} \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}_0} \boldsymbol{u}_k \leq \boldsymbol{b}, \\ & y_k \boldsymbol{h}_k(\boldsymbol{u}_k / y_k) \leq \boldsymbol{0}, \qquad k \in \mathcal{K}_0, \\ & \boldsymbol{y} \geq \boldsymbol{0}. \end{aligned}$$

The convex problem is in general a convex relaxation of (GLP), which has the same optimal value as (GLP) if one of the following regularity conditions is satisfied: (**i**) $\mathcal{X}_k$ is bounded for each $k \in \mathcal{K}_0$

(Gorissen et al., 2014, Lemma 1); (**ii**) there exists a $(\boldsymbol{y}, \{\boldsymbol{x}_k\}_k)$ with $\boldsymbol{y} > \boldsymbol{0}$ that is feasible for (GLP) (Zhen et al., 2023, Lemma 6). While for a special case where $\mathcal{X}_k$, $k \in \mathcal{K}$, are (nonempty) boxes, the corresponding linear relaxation of (GLP) is exact due to Dantzig (1963).

7.3. Convex hull representation for 0-1 mixed-integer convex programs

Consider the following 0-1 mixed-integer constrained set (Sherali and Adams, 2009):

$$\mathcal{X} = \left\{ \boldsymbol{x} \;\middle|\; \begin{array}{ll} h_k(\boldsymbol{x}) \leq 0, & k \in \mathcal{K}, \\ \boldsymbol{x}_B \in \{0,1\}^{n_B}, & \boldsymbol{x}_C \in [0,1]^{n_C} \end{array} \right\},$$

where the vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the concatenation of the binary vector $\boldsymbol{x}_B$ and the continuous vector $\boldsymbol{x}_C$, namely, $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_C)$ with $n_x = n_B + n_C$. Assume $\mathcal{X} \neq \emptyset$. Similarly to Sherali and Adams (2009), for each subset $\mathcal{J} \subseteq \mathcal{B}$, where $\mathcal{B} = \{1, \cdots, n_B\}$, we define the polynomial factors

$$F_{\mathcal{J}}(\boldsymbol{x}_B) = \prod_{j \in \mathcal{J}} (\boldsymbol{x}_B)_j \prod_{j \in \mathcal{B} \setminus \mathcal{J}} \big(1 - (\boldsymbol{x}_B)_j\big).$$

Observe that these factors satisfy the identity

$$\sum_{\mathcal{J} \subseteq \mathcal{B}} F_{\mathcal{J}}(\boldsymbol{x}_B) = \prod_{j \in \mathcal{B}} \big((\boldsymbol{x}_B)_j + (1 - (\boldsymbol{x}_B)_j)\big) = 1.$$

We linearize each factor $F_{\mathcal{J}}(\boldsymbol{x}_B)$ by introducing a variable $\lambda_{\mathcal{J}} \in \mathbb{R}_+$. We then impose the simplex constraints

$$\lambda_{\mathcal{J}} \geq 0 \quad \forall \mathcal{J} \subseteq \mathcal{B}, \qquad \sum_{\mathcal{J} \subseteq \mathcal{B}} \lambda_{\mathcal{J}} = 1, \tag{31}$$

and

$$(\boldsymbol{x}_B)_i = \sum_{\mathcal{J} \subseteq \mathcal{B} : i \in \mathcal{J}} \lambda_{\mathcal{J}} \quad \forall i \in \mathcal{B}. \tag{32}$$

Note that, since $\boldsymbol{x}_B \in \{0,1\}^{n_B}$, constraints (31) and (32) imply that

$$\lambda_{\mathcal{J}} = \begin{cases} 1, & \text{if } \mathcal{J} = \{i \in \mathcal{B} : (\boldsymbol{x}_B)_i = 1\}, \\ 0, & \text{otherwise.} \end{cases}$$

Hence, $\lambda_{\mathcal{J}} = F_{\mathcal{J}}(\boldsymbol{x}_B)$ for all $\mathcal{J} \subseteq \mathcal{B}$.

Next, for each subset $\mathcal{J} \subseteq \mathcal{B}$, we introduce a vector $\boldsymbol{y}_{\mathcal{J}} \in \mathbb{R}^{n_x}$ to represent the product $\boldsymbol{x}\lambda_{\mathcal{J}}$ componentwise. For the binary block, this yields the exact identities

$$\big((\boldsymbol{y}_{\mathcal{J}})_B\big)_i = \begin{cases} \lambda_{\mathcal{J}}, & \text{if } i \in \mathcal{J}, \\ 0, & \text{otherwise,} \end{cases} \qquad \forall i \in \mathcal{B},\ \forall \mathcal{J} \subseteq \mathcal{B}. \tag{33}$$

For the continuous block, we linearize the product $\boldsymbol{x}_C \lambda_{\mathcal{J}}$ by $(\boldsymbol{y}_{\mathcal{J}})_C$, and use the bounds $\boldsymbol{0} \leq \boldsymbol{x}_C \leq \boldsymbol{1}$ to impose

$$\boldsymbol{0} \leq (\boldsymbol{y}_{\mathcal{J}})_C \leq \boldsymbol{1}\lambda_{\mathcal{J}} \qquad \forall \mathcal{J} \subseteq \mathcal{B}, \tag{34}$$

where $\mathbf{0}, \mathbf{1} \in \mathbb{R}^{n_C}$ denote the all-zeros and all-ones vectors, respectively. Finally, since $\sum_{\mathcal{J}\subseteq\mathcal{B}} \lambda_{\mathcal{J}} = 1$ and $\boldsymbol{y}_{\mathcal{J}} = \boldsymbol{x}\lambda_{\mathcal{J}}$, we obtain

$$\boldsymbol{x} = \sum_{\mathcal{J}\subseteq\mathcal{B}} \boldsymbol{y}_{\mathcal{J}}. \tag{35}$$

Moreover, (33) and (35) imply (32), so the latter need not be imposed again.

Using these variables, and convexifying each bilinear constraint $\lambda_{\mathcal{J}} h_k(\boldsymbol{x}) \le 0$ via the perspective of h_k, we obtain

$$\mathcal{X}_{\text{conv}} = \left\{ \boldsymbol{x} \;\middle|\; \exists \boldsymbol{\lambda}, \boldsymbol{y}_{\mathcal{J}} : \begin{array}{l} \lambda_{\mathcal{J}} h_k\left(\dfrac{\boldsymbol{y}_{\mathcal{J}}}{\lambda_{\mathcal{J}}}\right) \le 0,\ k \in \mathcal{K},\ \mathcal{J} \subseteq \mathcal{B} \\ (31),\ (33),\ (34),\ (35) \end{array} \right\}. \tag{36}$$

The formulation (36) is convex and, as shown in Sherali and Adams (2009), it equals the convex hull of $\mathcal{X}$, that is,

$$\mathcal{X}_{\text{conv}} = \text{conv}(\mathcal{X}).$$

In particular, this construction enumerates all 2^{n_B} subsets $\mathcal{J} \subseteq \mathcal{B}$, corresponding to all binary assignments. Therefore, no branch and bound is required once all such subsets are included.

7.4. Approximate $\mathcal{S}$-Lemma for quadratically constrained quadratic optimization

Consider a quadratically constrained quadratic optimization problem with only one (quadratic) constraint:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{A}_0 \boldsymbol{x} + 2\boldsymbol{b}_0^\top \boldsymbol{x} + c_0 \\ \text{s.t.} & \boldsymbol{x}^\top \boldsymbol{A}_1 \boldsymbol{x} + 2\boldsymbol{b}_1^\top \boldsymbol{x} + c_1 \le 0, \end{array} \tag{QCQP}$$

where $\boldsymbol{A}_k \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{b}_k \in \mathbb{R}^{n_x}$ and $c_k \in \mathbb{R}$ for each $k \in \{0,1\}$. It is well-known that such a problem admits a convex reformulation via the $\mathcal{S}$-lemma. In the following, we show that the dual of the obtained convex reformulation from the $\mathcal{S}$-lemma can be interpreted as an RPT relaxation. Suppose that there exists an $\boldsymbol{x} \in \mathbb{R}^{n_x}$ with $\boldsymbol{x}^\top \boldsymbol{A}_1 \boldsymbol{x} + 2\boldsymbol{b}_1^\top \boldsymbol{x} + c_1 < 0$, then we have

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{A}_0 \boldsymbol{x} + 2\boldsymbol{b}_0^\top \boldsymbol{x} + c_0 \\ \text{s.t.} & \boldsymbol{x}^\top \boldsymbol{A}_1 \boldsymbol{x} + 2\boldsymbol{b}_1^\top \boldsymbol{x} + c_1 \le 0 \end{array} \iff \begin{array}{ll} \max\limits_{\lambda \ge 0, \gamma} & \gamma \\ \text{s.t.} & \begin{bmatrix} \boldsymbol{A}_0 & \frac{1}{2}\boldsymbol{b}_0 \\ \frac{1}{2}\boldsymbol{b}_0^\top & c_0 \end{bmatrix} \succeq \gamma \begin{bmatrix} \boldsymbol{O} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} - \lambda \begin{bmatrix} \boldsymbol{A}_1 & \frac{1}{2}\boldsymbol{b}_1 \\ \frac{1}{2}\boldsymbol{b}_1^\top & c_1 \end{bmatrix}, \end{array}$$

where $\boldsymbol{O} \in \mathbb{R}^{n_x \times n_x}$ is a matrix of all zeros. Here the " $\iff$ " holds due to the $\mathcal{S}$-lemma (Boyd and Vandenberghe, 2004, Appendix B). The dual of the obtained semidefinite problem is

$$\begin{array}{ll} \min\limits_{\boldsymbol{X}, \boldsymbol{x}} & \text{Tr}\,(\boldsymbol{A}_0 \boldsymbol{X}) + 2\boldsymbol{b}_0^\top \boldsymbol{x} + c_0 \\ \text{s.t.} & \text{Tr}\,(\boldsymbol{A}_1 \boldsymbol{X}) + 2\boldsymbol{b}_1^\top \boldsymbol{x} + c_1 \le 0, \\ & \begin{bmatrix} \boldsymbol{X} & \boldsymbol{x} \\ \boldsymbol{x}^\top & 1 \end{bmatrix} \succeq 0, \end{array}$$

which is clearly an RPT relaxation of (QCQP). Consider now a generic quadratically constrained quadratic optimization problem with more than one quadratic inequality constraint:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{A}_0 \boldsymbol{x} + 2\boldsymbol{b}_0^\top \boldsymbol{x} + c_0 \\ \text{s.t.} & \boldsymbol{x}^\top \boldsymbol{A}_k \boldsymbol{x} + 2\boldsymbol{b}_k^\top \boldsymbol{x} + c_k \leq 0 \quad k \in \mathcal{K}, \end{array}$$

where $\boldsymbol{A}_k \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{b}_k \in \mathbb{R}^{n_x}$ and $c_k \in \mathbb{R}$ for each $k \in \mathcal{K}_0$. Similarly, the dual of the convex relaxation obtained from using the approximate $\mathcal{S}$-lemma coincides with the convex relaxation from RPT:

$$\begin{array}{ll} \min\limits_{\boldsymbol{X},\boldsymbol{x}} & \operatorname{Tr}(\boldsymbol{A}_0 \boldsymbol{X}) + 2\boldsymbol{b}_0^\top \boldsymbol{x} + c_0 \\ \text{s.t.} & \operatorname{Tr}(\boldsymbol{A}_k \boldsymbol{X}) + 2\boldsymbol{b}_k^\top \boldsymbol{x} + c_k \leq 0, \quad k \in \mathcal{K}, \\ & \begin{bmatrix} \boldsymbol{X} & \boldsymbol{x} \\ \boldsymbol{x}^\top & 1 \end{bmatrix} \succeq 0. \end{array}$$

Note that here the obtained relaxation is not tight in general, and for more details on the approximate $\mathcal{S}$-lemma, we refer to Ben-Tal et al. (2002).

7.5. Fractional optimization

Consider the following generic fractional optimization problem

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \dfrac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \\ \text{s.t.} & h_k(\boldsymbol{x}) \leq 0 \quad k \in \mathcal{K}, \end{array} \tag{FP}$$

where $f: \mathbb{R}^{n_x} \to \mathbb{R}_+$ is convex and nonnegative, $g: \mathbb{R}^{n_x} \to \mathbb{R}_{++}$ is concave and positive, and $h_k: \mathbb{R}^{n_x} \to (-\infty, +\infty]$ is proper, closed and convex for every $k \in \mathcal{K}$. By first introducing an epigraphical variable τ for the positive convex function $1/g(\boldsymbol{x})$, we obtain the SLC constraint $\tau g(\boldsymbol{x}) \geq 1$, and then apply RPT to obtain:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x},\tau} & \tau f(\boldsymbol{v}/\tau) \\ \text{s.t.} & \tau g(\boldsymbol{v}/\tau) \geq 1, \\ & \tau h_k(\boldsymbol{v}/\tau) \leq 0, \quad k \in \mathcal{K}. \end{array}$$

The obtained convex problem is an exact convex reformulation of (FP) (Schaible, 1974).

8. Numerical experiments

In this section, we demonstrate the efficiency and effectiveness of our RPT-based branch-and-bound methods on several nonconvex optimization problems, including a sum-of-max-linear-terms maximization problem, a log-sum-exp maximization problem, a linear multiplicative optimization problem, a quadratically constrained quadratic optimization problem, and a dike height optimization problem. Using the biconjugate reformulation, we show that the first three problems can be formulated as bilinear optimization problems subject to convex and nonconvex, though SLC, constraints. The latter two problems are already in generic form (1). In the implementation of RPT-BB, all problems are assumed to be minimization problems, by switching to the minus of the objective if necessary. Moreover, in all problems that we address, except for the linear multiplicative optimization problem, the conditions for convergence of RPT-BB are satisfied.

Numerical experiments were performed on a MacBook Pro with an Apple M3 Pro chip (11 CPU cores: 5 performance and 6 efficiency) and 18 GB RAM. All computations for RPT-BB, RPT-SDP-BB, and SCIP were implemented in Julia 1.12.5 (JuliaLang contributors, 2026) using JuMP.jl 1.34.3 (JuMP developers, 2026), with MOSEK version 11.0.30 (ApS, 2025), Gurobi version 13.0.0 (Gurobi Optimization, LLC, 2026), and SCIP version 10.0.0 (Hojny et al., 2025). All computations were carried out with BARON version 21.1.13 (Sahinidis, 1996; The Optimization Firm, LLC, 2021), using Pyomo version 6.7.2 as the modeling interface (Bynum et al., 2021; Hart et al., 2011). Finally, all computations with CPLEX were carried out with CPLEX version 22.1.2 (ILOG, Inc., 2017) and implemented using the Python package `docplex` version 2.23.259. Within RPT-BB and RPT-SDP-BB, we use Gurobi for the linear optimization problems and MOSEK for the nonlinear optimization problems. In all branch and bound implementations the optimality gap δ is set to 10^{-4}. We compare our approach, when applied with (RPT-SDP-BB) and without the LMI (RPT-BB), with BARON, and with SCIP, applied either on the direct formulation (BARON-Dir, SCIP-Dir) or the biconjugate reformulation (BARON-Bic, SCIP-Bic), where applicable.

In the remainder of this section, in the tables depicting the results, we use the abbreviations “Opt”, “Hyp”, “SHyp”, and “Time” to represent the optimal value, the total number of hyperplanes generated during branch and bound, the number of secant hyperplanes, and the computation time in seconds, respectively. Moreover, we set the maximum time limit equal to 1800 seconds, hence if the computation time equals 1800^*, the optimum cannot be found within 1800 seconds and all approaches return the best value they can obtain within 1800 seconds. A superscript k on a computation time indicates that k out of 10 randomly generated instances reached the time limit without proving optimality; in such cases, we use the incumbent objective value at termination and record the runtime as 1800 seconds. Finally, a “-” indicates that no solution was returned after 1800 seconds.

8.1. Sum-of-max-linear-terms maximization over convex and nonconvex constraints

We consider the following generic sum-of-max-linear-terms maximization problem from Zhen et al. (2022) and Selvi et al. (2022):

$$\max_{\boldsymbol{x}\in\mathcal{X}} \quad \sum_{\ell\in\mathcal{L}} \max_{j\in\mathcal{J}_\ell}\{\boldsymbol{A}_j^\top \boldsymbol{x} + b_j\}, \tag{37}$$

where $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_y}$, $\boldsymbol{b} \in \mathbb{R}^{n_y}$. We consider three cases of $\mathcal{X}$: a set defined by linear constraints, a set defined by an additional geometric constraint, and a set defined by an additional nonconvex constraint, i.e., $\mathcal{X} = \mathcal{X}_1$, $\mathcal{X} = \mathcal{X}_2$, and $\mathcal{X} = \mathcal{X}_3$, where

$$\begin{aligned}
\mathcal{X}_1 &= \{\boldsymbol{x} \in \mathbb{R}^{n_x}_+ \mid \boldsymbol{D}^\top \boldsymbol{x} \leq \boldsymbol{d}\}, \\
\mathcal{X}_2 &= \left\{\boldsymbol{x} \in \mathcal{X}_1 \;\middle|\; \log\left(\sum_{i=1}^{n_x} \exp(x_i)\right) \leq a\right\}, \\
\mathcal{X}_3 &= \left\{\boldsymbol{x} \in \mathcal{X}_1 \;\middle|\; \|\boldsymbol{x}\|_2 + \sum_{i=1}^{n_x} \sqrt{x_i} \leq c\right\}.
\end{aligned}$$

Here $\boldsymbol{D} \in \mathbb{R}^{n_x \times m}$ and $\boldsymbol{d} \in \mathbb{R}^m$. Since the objective of (37) is a closed convex function, we can replace it by its biconjugate function and obtain the following equivalent maximization problem:

$$\max_{\substack{\boldsymbol{x} \in \mathcal{X} \\ \boldsymbol{y} \in \mathcal{Y}}} \quad (\boldsymbol{A}^\top \boldsymbol{x} + \boldsymbol{b})^\top \boldsymbol{y}, \tag{38}$$

where $\mathcal{Y}$ equals the domain of the conjugate function of the objective of (37), i.e.,

$$\mathcal{Y} = \left\{\boldsymbol{y} \in \mathbb{R}^{n_y}_+ \;\middle|\; \sum_{j \in \mathcal{J}_\ell} y_j = 1, \ell \in \mathcal{L}\right\}.$$

Note that $n_y = \sum_{\ell \in \mathcal{L}} |\mathcal{J}_\ell|$. We compare RPT-BB, RPT-SDP-BB, and BARON. Furthermore, for $\mathcal{X} = \mathcal{X}_1$ and $\mathcal{X} = \mathcal{X}_2$, we also compare them with the exact mixed integer optimization reformulation (MIP), given by

$$\begin{array}{lll}
\max\limits_{\substack{\boldsymbol{x} \in \mathcal{X} \\ \boldsymbol{\lambda}, \boldsymbol{z}}} & \sum\limits_{\ell \in \mathcal{L}} \lambda_\ell & \\
\text{s.t.} & \lambda_\ell \leq \boldsymbol{A}_j^\top \boldsymbol{x} + b_j + M(1 - z_j), & j \in \mathcal{J}_\ell, \ell \in \mathcal{L}, \\
& \sum\limits_{j \in \mathcal{J}_\ell} z_j = 1, & \ell \in \mathcal{L}, \\
& \boldsymbol{z} \in \{0,1\}^{n_y}. &
\end{array} \tag{39}$$

We solve Problem (39) with Gurobi for $\mathcal{X}_1$ and MOSEK for $\mathcal{X}_2$. Instances labeled 1–5 correspond to single instances, taken from Selvi et al. (2022), whereas instances labeled 1r–5r report averages over 10 independently generated random instances, generated according to the exact same mechanism. We refer to Appendices D.1 and E.1 for the convex RPT relaxation and problem instances, respectively.

From Table 4, we observe that for $\mathcal{X} = \mathcal{X}_1$, all methods find the global optimum for every instance. MIP achieves the lowest computation time for instances 1–4 and their averaged counterparts 1r–4r, solving them in at most 1.18 seconds. For instances 5 and 5r, RPT-BB has the lowest computation time (1.81 and 11.00 seconds, respectively), whereas MIP requires significantly more time and BARON cannot prove optimality within the time limit. We further observe that for instances 3 and 3r, RPT-SDP-BB achieves a lower computation time than RPT-BB, which can be attributed to the tighter relaxation provided by the semidefinite constraint.

$\mathcal{X}$	#	RPT-BB				RPT-SDP-BB				BARON		MIP	
		Opt	Time	Hyp	SHyp	Opt	Time	Hyp	SHyp	Opt	Time	Opt	Time
$\mathcal{X}_1$	1	23.29	0.01	0.0	0.0	23.29	0.02	0.0	0.0	23.29	0.16	23.29	0.00
	1r	22.72	0.01	0.0	0.0	22.72	0.02	0.0	0.0	22.72	0.10	22.72	0.00
	2	233.94	0.09	0.0	0.0	233.94	0.61	0.0	0.0	233.94	0.24	233.94	0.02
	2r	211.87	0.19	0.2	0.0	211.87	0.51	0.0	0.0	211.87	0.77	211.87	0.02
	3	1081.62	115.15	73.0	48.0	1081.62	32.44	0.0	0.0	1081.62	1800.00*	1081.62	1.18
	3r	1159.90	51.95	37.0	25.6	1159.90	48.63	0.3	0.3	1159.90	1800.00*	1159.90	1.06
	4	113.70	0.02	0.0	0.0	113.70	0.09	0.0	0.0	113.70	2.12	113.70	0.01
	4r	83.78	0.09	0.4	0.4	83.78	0.06	0.0	0.0	83.78	2.67	83.78	0.00
	5	3002.44	1.81	1.0	1.0	3002.44	71.63	1.0	1.0	3002.44	1800.00*	3002.44	327.52
	5r	2898.05	11.00	6.5	5.1	2898.05	73.41	1.1	1.1	2898.05	1800.00*	2898.05	1073.88[4]
$\mathcal{X}_2$	1	14.54	0.02	0.0	0.0	14.58	0.03	0.0	0.0	14.58	0.75	14.56	0.05
	1r	14.54	0.02	0.0	0.0	14.54	0.03	0.0	0.0	14.54	0.13	14.54	0.06
	2	136.22	0.38	0.0	0.0	136.22	1.05	0.0	0.0	136.22	1800.00*	136.22	1115.66
	2r	122.21	2.68	3.7	2.1	122.21	2.54	0.7	0.7	122.21	1800.00*	122.21	468.78
	3	837.94	325.78	104.0	79.0	837.94	94.39	0.0	0.0	837.94	1800.00*	790.75	1800.00*
	3r	890.07	765.24	233.2	192.2	890.07	231.52	0.6	0.6	890.07	1800.00*	843.66	1800.00*
	4	33.73	4.12	16.0	12.0	33.73	2.00	2.0	0.0	33.73	43.00	33.73	0.13
	4r	31.81	1.43	5.8	4.5	31.81	0.70	0.8	0.0	31.81	20.19	31.81	0.11
	5	1610.69	1800.00*	561.0	88.0	1610.69	1800.00*	11.0	11.0	1610.69	1800.00*	1588.59	1800.00*
	5r	1670.92	1800.00*	563.0	133.5	1670.92	1800.00*	11.0	11.0	1671.02	1800.00*	1657.09	1800.00*
$\mathcal{X}_3$	1	13.45	107.81	676.0	636.0	13.44	5.01	10.0	10.0	13.45	0.29		
	1r	15.02	346.06[1]	439.0	351.1	14.99	11.88	18.7	18.7	15.02	0.14		
	2	140.88	59.61	138.0	137.0	140.62	28.73	4.0	4.0	140.89	5.22		
	2r	129.29	95.89	213.0	208.1	129.08	26.74	3.8	3.8	129.31	231.13		
	3	768.96	264.37	48.0	45.0	768.96	1800.00*	5.0	5.0	768.97	1800.00*		
	3r	805.98	587.95[2]	70.9	70.8	805.96	1733.86[6]	3.9	3.9	805.60	1800.00*		
	4	45.34	45.08	84.0	83.0	45.34	147.34	58.0	58.0	45.34	74.28		
	4r	44.78	1137.76[5]	569.3	555.6	44.86	310.24[1]	65.2	65.2	44.97	40.02		
	5	2268.17	1800.00*	245.9	77.0	-	1800.00*	5.0	5.0	2264.37	1800.00*		
	5r	2359.75	1800.00*	245.9	66.3	-	1800.00*	5.2	5.2	2186.57	1800.00*		

Table 4 Results for Problem (38) over the feasible regions $\mathcal{X}_1, \mathcal{X}_2$, and $\mathcal{X}_3$.

For $\mathcal{X} = \mathcal{X}_2$, RPT-BB achieves the lowest computation time for instances 1, 1r, and 2, while MIP is fastest for instances 4 and 4r. For instance 2r, RPT-BB and RPT-SDP-BB achieve comparable computation times (2.68 and 2.54 seconds, respectively). For instances 3 and 3r, RPT-SDP-BB achieves the lowest computation time, solving them in 94.39 and 231.52 seconds, respectively, whereas BARON cannot prove optimality within the time limit. Moreover, for instances 3, 3r, 5, and 5r, MIP does not find the global optimum within the time limit. For instances 5 and 5r, all methods except MIP find the same best objective value but cannot prove optimality within the time limit.

For $\mathcal{X} = \mathcal{X}_3$, the MIP reformulation is not applicable due to the nonconvex constraint. BARON achieves the lowest computation time for instances 1, 1r, 2, and 4r, while RPT-SDP-BB is fastest for instance 2r and RPT-BB for instance 4. Moreover, RPT-BB is the only method that proves global optimality for instance 3 within the time limit, and for instance 3r, RPT-BB solves the most instances to optimality (8 out of 10). For instances 5 and 5r, none of the methods can prove optimality, but RPT-BB obtains better objective values than BARON, while RPT-SDP-BB does not find a feasible solution. We refer to Appendix A for a comparison of three branching strategies used within RPT-BB and RPT-SDP-BB.

8.2. Log-sum-exp maximization over linear constraints

We consider the log-sum-exp maximization problem subject to linear constraints:

$$\max_{\boldsymbol{x}\in\mathcal{X}_1} \quad \log\left(\sum_{i=1}^{n_x}\exp(x_i)\right). \tag{40}$$

Since the objective of (40) is a closed convex function, we can replace it by its biconjugate function and obtain the following equivalent maximization problem:

$$\max_{\substack{\boldsymbol{x}\in\mathcal{X}_1\\(\boldsymbol{y},\boldsymbol{w})\in\mathcal{Y}}} \quad \boldsymbol{x}^\top\boldsymbol{y}+\sum_{i=1}^{n_x} w_i, \tag{41}$$

where $\mathcal{Y}$ equals the domain of the conjugate function of the objective of (40), i.e.,

$$\mathcal{Y}=\left\{\boldsymbol{y}\in\mathbb{R}^{n_x}_+,\ \boldsymbol{w}\in\mathbb{R}^{n_x}\ \middle|\ y_i\exp\left(\frac{w_i}{y_i}\right)\le 1,\ i\in\{1,\dots,n_x\},\ \sum_{i=1}^{n_x} y_i=1\right\}.$$

Observe that here we make use of case 7 in Table 1 and introduce epigraphical variables w_i for every $i\in\{1,\dots,n_x\}$. We refer to Appendices D.2 and E.2 for the convex RPT relaxation and problem instances, respectively. The results are illustrated in Table 5. Each entry corresponds to the average over 10 randomly generated instances.

$\mathcal{X}$	#	RPT-BB				RPT-SDP-BB				BARON-Dir		SCIP-Dir		BARON-Bic		SCIP-Bic	
		Opt	Time	Hyp	SHyp	Opt	Time	Hyp	SHyp	Opt	Time	Opt	Time	Opt	Time	Opt	Time
	1	10.01	0.06	0.0	0.0	10.01	0.20	0.0	0.0	10.01	0.07	10.01	0.00	10.01	1800.00*	10.01	1800.00*
	2	40.00	1.28	0.0	0.0	40.00	32.92	0.0	0.0	40.00	0.19	40.00	0.00	33.34	1800.00*	-	1800.00*
$\mathcal{X}_1$	3	6.10	1.71	1.0	0.9	6.10	2.20	0.8	0.8	6.10	0.11	6.10	0.03	5.32	1800.00*	6.10	1800.00*
	4	21.96	0.31	0.0	0.0	21.96	1.68	0.0	0.0	21.90	360.93[2]	21.96	0.16	-	1800.00*	20.89	1800.00*
	5	34.76	5.47	0.0	0.0	34.76	124.45	0.0	0.0	34.76	360.96[2]	34.76	0.28	-	1800.00*	33.40	1800.00*

Table 5 Results for Problem (40).

From Table 5, we observe that both RPT-BB and RPT-SDP-BB find the global optimum for all instances. For instances 1–3, SCIP-Dir and BARON-Dir achieve the lowest computation times, solving them in fractions of a second. For instances 4 and 5, SCIP-Dir remains the fastest, while BARON-Dir encounters timeouts on 2 out of 10 randomly generated instances and does not always find the global optimum. RPT-BB solves all instances, achieving computation times of 0.31 and 5.47 seconds for instances 4 and 5, respectively, and outperforms RPT-SDP-BB in all instances. We further observe that BARON-Bic and SCIP-Bic fail to find the global optimum within the time limit for most instances, and do not return any feasible solution for instances 4–5 (BARON-Bic) and instance 2 (SCIP-Bic), demonstrating that the biconjugate reformulation (41) is significantly harder for general-purpose solvers. We refer to Appendix A for a comparison of three branching strategies.

8.3. Linear multiplicative optimization

We consider the following linear multiplicative optimization problem from Ryoo and Sahinidis (1996):

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad \prod_{i=1}^{n_y} \boldsymbol{A}_i^\top \boldsymbol{x} + b_i, \tag{42}$$

where $\boldsymbol{A}\in\mathbb{R}^{n_x\times n_y}$, $\boldsymbol{b}\in\mathbb{R}^{n_y}$, $\mathcal{X}=\{\boldsymbol{x}\in\mathbb{R}^{n_x}_+ \mid \boldsymbol{D}^\top\boldsymbol{x}\leq\boldsymbol{d},\ \boldsymbol{A}_i^\top\boldsymbol{x}+b_i\geq 0\}$, $\boldsymbol{D}\in\mathbb{R}^{n_x\times L'}$, and $\boldsymbol{d}\in\mathbb{R}^{L'}$. Without loss of generality we assume $\boldsymbol{A}_i^\top\boldsymbol{x}+b_i>0$ for all $i\in\mathcal{I}$. Utilizing a log transformation, as in Ryoo and Sahinidis (1996), Problem (42) can be equivalently reformulated as

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad \sum_{i=1}^{n_y} \log\left(\boldsymbol{A}_i^\top\boldsymbol{x}+b_i\right). \tag{43}$$

Since the objective of (43) is a closed concave function, we can replace it by its biconjugate function and obtain the following equivalent maximization problem:

$$\max_{\substack{\boldsymbol{x}\in\mathcal{X}\\(\boldsymbol{y},\boldsymbol{w})\in\mathcal{Y}}} \quad -(\boldsymbol{A}^\top\boldsymbol{x}+\boldsymbol{b})^\top\boldsymbol{y}-\sum_{i=1}^{n_y} w_i, \tag{44}$$

where $\mathcal{Y}$ equals the domain of the conjugate function of the objective of (43), i.e.,

$$\mathcal{Y}=\left\{\boldsymbol{y}\in\mathbb{R}^{n_y}_+, \boldsymbol{w}\in\mathbb{R}^{n_y} \;\middle|\; \exp(-w_i-1)\leq y_i,\ i\in\{1,\ldots,n_y\}\right\}.$$

Observe that here we make use of case 3 in Table 1 and introduce epigraphical variables w_i for every $i\in\{1,\ldots,n_y\}$. Moreover, observe that $\mathcal{Y}$ is not bounded. Hence, the conditions for convergence of RPT-BB are not satisfied. We refer to Appendices D.3 and E.3 for the convex RPT relaxation and problem instances, respectively. The results are illustrated in Table 6. Each entry corresponds to the average over 10 randomly generated instances.

$\mathcal{X}$	#	RPT-BB				RPT-SDP-BB				BARON-Dir		SCIP-Dir		BARON-Bic		SCIP-Bic	
		Opt	Time	Hyp	SHyp	Opt	Time	Hyp	SHyp	Opt	Time	Opt	Time	Opt	Time	Opt	Time
	1	12.14	1.24	25.6	2.2	12.14	6.74	20.7	0.3	12.14	0.27	12.14	2.12	12.14	552.27^{3}	12.14	3.64
	2	23.47	6.33	56.2	18.9	23.47	18.15	30.3	5.7	23.47	1800.00*	23.47	31.14	23.47	1664.19^{9}	23.47	800.19^{2}
	3	20.96	26.19	151.5	79.7	20.96	21.14	25.1	2.2	20.96	1261.47^{7}	20.96	36.87	20.96	1800.00*	20.96	1283.12^{6}
	4	18.99	1590.08^{8}	1778.4	1442.7	18.99	84.61	35.7	15.4	18.99	1584.35^{8}	18.99	570.33^{2}	18.99	1800.00*	18.99	1800.04*
	5	8.90	1800.00*	364.5	96.7	8.90	536.41^{1}	40.8	17.5	8.90	236.84^{1}	8.90	4.39	8.90	845.89^{4}	8.90	132.23

Table 6 Results for Problem (42).

From Table 6, we observe that our approach often outperforms BARON and SCIP on linear multiplicative optimization problems. Specifically, for instance 1, BARON-Dir achieves the lowest computation time (0.27 seconds), while for instance 5, SCIP-Dir is the fastest (4.39 seconds). For instance 2, which involves 10 linear terms, RPT-BB achieves the lowest computation time (6.33 seconds). For instances 3 and 4, which involve 9 and 8 linear terms, respectively, RPT-SDP-BB achieves the lowest computation time (21.14 and 84.61 seconds, respectively), while RPT-BB and BARON-Dir both encounter frequent timeouts for instance 4. Moreover, BARON-Bic encounters timeouts for all instances, while SCIP-Bic encounters timeouts for instances 2–4. We further observe that RPT-SDP-BB consistently achieves lower computation times than RPT-BB for instances 3–5, which can be attributed to the tighter relaxation reducing the number of hyperplanes needed during branch and bound. We refer to Appendix A for a comparison of three branching strategies.

8.4. Quadratically constrained quadratic optimization

We consider the following quadratically constrained quadratic optimization problem:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}_1} \quad & \boldsymbol{x}^\top \boldsymbol{P}_0 \boldsymbol{x} + \boldsymbol{q}_0^\top \boldsymbol{x} + r_0 \\ \text{s.t.} \quad & \boldsymbol{x}^\top \boldsymbol{P}_k \boldsymbol{x} + \boldsymbol{q}_k^\top \boldsymbol{x} + r_k \le 0, \qquad k \in \mathcal{K}, \end{aligned} \tag{45}$$

where $\boldsymbol{P}_k \in \mathbb{R}^{n_x \times n_x}$, $k \in \mathcal{K}_0$, are not necessarily positive semidefinite and as a result Problem (45) is not necessarily convex. However, the nonconvex quadratic functions are SLC, hence we can apply RPT. In the first five instances involving nonconvex QPs over linear constraints, we also compare with CPLEX, which has a specialized algorithm for these problems. We refer to Appendices D.4 and E.4 for the convex RPT relaxation and problem instances, respectively. The results are illustrated in Table 7.

$\mathcal{X}$	#	RPT-BB				RPT-SDP-BB				BARON		CPLEX	
		Opt	Time	Hyp	SHyp	Opt	Time	Hyp	SHyp	Opt	Time	Opt	Time
	1	394.75	1.00	1.0	1.0	394.75	1.12	1.0	1.0	394.75	0.38	394.75	0.11
	2	884.75	0.06	0.0	0.0	884.75	0.10	0.0	0.0	884.75	0.28	884.75	0.05
	3	6888.78	0.01	0.0	0.0	6888.78	0.01	0.0	0.0	6888.78	0.07	6888.78	0.01
	4	98382.63	2.41	1.0	0.0	98382.63	2.04	0.0	0.0	98382.63	1800.00*	98382.63	0.67
	5	774482.39	87.03	1.0	0.0	774482.39	43.34	0.0	0.0	774482.38	1800.00*	774482.38	7.53
	6	1717.80	0.04	1.0	0.0	1717.80	0.02	0.0	0.0	1717.80	0.14		
	7	4507.75	0.06	2.0	2.0	4507.75	0.13	2.0	2.0	4507.75	0.19		
	8	17084.82	0.17	3.0	2.0	17084.82	0.11	1.0	1.0	17084.82	0.17		
	9	46631.07	955.72	4330.0	2581.0	-	1800.00*	406.0	325.0	46632.36	17.44		
	10	-	1800.00*	3667.0	1680.0	52233.41	81.69	46.0	46.0	52233.70	1800.00*		

Table 7 Results for Problem (45). Instances 1–5 contain only linear constraints, while instances 6–10 contain additional nonconvex quadratic constraints.

From Table 7, we observe that for instances 1–3, all approaches find the optimum within approximately one second. CPLEX achieves the lowest computation time for instances 4 and 5 (0.67 and 7.53 seconds, respectively), while BARON cannot prove optimality within the time limit for these instances. RPT-BB and RPT-SDP-BB also solve instances 4 and 5, with RPT-SDP-BB being faster than RPT-BB. For instances 6–8, which contain additional nonconvex quadratic constraints, RPT-BB and RPT-SDP-BB achieve the lowest computation times, solving all instances in less than 0.2 seconds. For instance 9, BARON achieves the lowest computation time (17.44 seconds), while

RPT-BB also finds the global optimum but requires 955.72 seconds, and RPT-SDP-BB does not find a feasible solution within the time limit. For instance 10, RPT-SDP-BB achieves the lowest computation time (81.69 seconds), whereas BARON finds the optimum but cannot prove optimality, and RPT-BB does not find a feasible solution within the time limit. We refer to Appendix A for a comparison of three branching strategies.

8.5. Dike height optimization

Eijgenraam et al. (2017) develop a model to optimize dike heightening in the Netherlands. The authors show that the optimal solution is periodic, i.e., the dike is heightened by the same amount every t years, and explicit expressions are derived for t and the optimal heightenings. However, in practice there are several reasons to deviate from the periodic solution. For example, it may be desirable to combine heightenings of several dikes. In this section, we propose to use RPT to solve the dike heightening problem in which the years at which heightening takes place are fixed and may deviate from every t years. Such problems cannot be solved by the approach in Eijgenraam et al. (2017). We consider the following dike height optimization problem, which is the time truncated version of the problem in Eijgenraam et al. (2017):

$$\min_{\boldsymbol{x}\geq \mathbf{0},\boldsymbol{h}} \underbrace{\sum_{k\in\mathcal{K}_0}(C+bx_k)\exp(\lambda h_k-\delta t_k)}_{\text{Investment costs}}+\underbrace{\sum_{k\in\mathcal{K}_0}\frac{S_0}{\beta_\delta}\left(\exp(\beta_\delta t_{k+1})-\exp(\beta_\delta t_k)\right)\exp(-\theta h_k)}_{\text{Expected damage costs}}+\underbrace{\frac{S_0}{\delta}\exp(\beta_\delta T-\theta h_K)}_{\text{Future damage costs}},\tag{DHO}$$

where $\boldsymbol{t}$ is the vector of all moments in time at which the dike height is increased, $t_0=0$, $\boldsymbol{x}$ is the vector of all increases in dike height, where x_k is the increment of the dike height at time t_k, h_k is the increase in dike height after t_k years, i.e., $h_k=\sum_{i=0}^{k}x_i$, $h_K=\sum_{k\in\mathcal{K}_0}x_k$ and $\beta_\delta,\delta,\theta,\lambda,b,C,T$ and S_0 are constants, which are explained in more detail in Appendix E.5. Observe that the feasible region is not compact. However, we can add redundant upper bounds on $\boldsymbol{x}$ such that we obtain a compact feasible region. Moreover, since $C>0$, the conditions for convergence of RPT-BB, as specified in Theorem 3, are satisfied.

The objective of (DHO) is to minimize the sum of investment costs and the total expected cost of flooding, both as a result of heightening dikes, see Eijgenraam et al. (2017) for a full description. Since $\boldsymbol{t}$ is fixed, the objective of (DHO) is SLC, as it consists of two convex terms (expected damage costs and future damage costs) and a sum of linear times convex functions, hence we can apply RPT-BB. We compare RPT-BB, RPT-SDP-BB, and BARON. We refer to Appendices D.5 and E.5 for the convex RPT relaxation and problem instances, respectively. The results for the homogeneous dike rings 10, 15 and 16 in the Netherlands are shown in Table 8.

t	#	RPT-BB				RPT-SDP-BB				BARON	
		Opt	Time	Hyp	SHyp	Opt	Time	Hyp	SHyp	Opt	Time
t_{ir}	10	61.98	1800.00*	6499.0	2015.0	61.98	0.15	0	0	61.98	105.08
	15	608.74	1800.00*	7344.0	1494.0	608.74	0.16	0	0	608.74	1800.00*
	16	1286.11	1800.00*	7953.0	1903.0	1268.11	0.14	0	0	1268.11	1800.00*
t_{25}	10	61.31	1800.00*	4476.0	1773.0	61.31	0.68	0	0	61.31	1092.25
	15	609.92	1800.00*	4322.0	672.0	609.92	0.23	0	0	609.92	1800.00*
	16	1269.63	1800.00*	4485.0	846.0	1269.63	0.24	0	0	1269.63	1800.00*
t_{50}	10	55.50	580.42	3229.0	1060.0	55.50	0.06	0	0	55.50	1.62
	15	545.23	146.77	1078.0	191.0	545.23	0.07	0	0	545.23	1.12
	16	1100.07	71.62	616.0	380.0	1100.07	0.14	0	0	1100.07	1800.00*

Table 8 Results for Problem (DHO), for dike rings 10, 15, and 16 in the Netherlands.

From Table 8, we observe that RPT-SDP-BB solves every instance at the root node in less than a second, achieving the lowest computation time across all instances. RPT-BB finds the optimal solution for t_{50} (580.42, 146.77, and 71.62 seconds for dike rings 10, 15, and 16, respectively), but cannot prove optimality within the time limit for t_{ir} and t_{25}. BARON proves optimality for t_{50} on dike rings 10 and 15 (1.62 and 1.12 seconds, respectively) and for dike ring 10 on t_{ir} (105.08 seconds) and t_{25} (1092.25 seconds), but cannot prove optimality within the time limit for the remaining instances. We further observe that the number of hyperplanes generated by RPT-BB is substantially larger for t_{ir} and t_{25} than for t_{50}, indicating that the branch and bound procedure requires more refinement for the irregular and 25-year heightening schedules. We refer to Appendix A for a comparison of three branching strategies.

9. Discussion and conclusion

In summary, we develop a method for globally solving nonconvex optimization problems involving SLC functions. We introduce the RPT framework, which enables us to obtain a convex relaxation of the original nonconvex problem, while introducing additional variables and constraints. We then incorporate it in spatial branch and bound, combining the new RPT relaxation with a modification of eigenvector branching that also handles the indefinite case, in order to solve the initial problem to optimality by sequentially partitioning the feasible region into smaller regions. In the numerical experiments, we demonstrate that the proposed method compares favorably with current state-of-the-art global optimization methods. Overall, we observe that for the considered problem instances, RPT-BB and RPT-SDP-BB are able to solve most problems by generating a few branching hyperplanes. This, together with the efficiency of MOSEK for solving conic optimization problems, is what drives the speed of the method.

A key limitation of the RPT-BB method is its reduced tractability when applied to problems with a large number of variables and constraints. This issue arises because the method involves squaring the number of variables and performing pairwise multiplications across all linear and convex constraints within the feasible region. To enhance scalability, future research could explore methodological adaptations designed to manage larger problem instances more efficiently. Potential adaptations might include employing partial constraint multiplications and selectively generating variable products. An iterative constraint generation approach could also be beneficial. This would involve initially solving the RPT relaxation with a subset of constraints, then progressively incorporating additional constraints based on whether the solutions violate any remaining constraints. Assessing the efficiency of this iterative approach, considering it requires multiple resolutions of the RPT relaxation, would be essential.

Finally, it would be interesting to investigate the potential of RPT-BB beyond the scope discussed in this paper, by applying RPT-BB also to nonconvex optimization problems in other fields, such as mixed integer nonlinear optimization, robust optimization, adaptive robust optimization, distributionally robust optimization, polynomial optimization, and bilevel optimization.

References

T. Achterberg. Scip: solving constraint integer programs. *Mathematical Programming Computation*, 1:1–41, 2009.

F. Al-Khayyal, C. Larsen, and T. Van Voorhis. A relaxation method for nonconvex quadratically constrained quadratic programs. *Journal of Global Optimization*, 6(3):215–230, 1995.

K. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2):471–484, 2009.

K. Anstreicher. On convex relaxations for quadratically constrained quadratic programming. *Mathematical Programming*, 136(2):233–251, 2012.

K. Anstreicher. Kronecker product constraints with an application to the two-trust-region subproblem. *SIAM Journal on Optimization*, 27(1):368–378, 2017.

K. Anstreicher. Solving two-trust-region subproblems using semidefinite optimization with eigenvector branching. *Journal of Optimization Theory and Applications*, 202(1):303–319, 2022.

MOSEK ApS. *MOSEK Optimization Suite 11.0.30 Documentation*, 2025. URL https://docs.mosek.com/11.0/capi/intro_info.html. MOSEK v11.0.30 used in experiments.

X. Bao, N. Sahinidis, and M. Tawarmalani. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, 129:129–157, 2011.

A. Ben-Tal and E. Roos. An algorithm for maximizing a convex function based on its minimum. *INFORMS Journal on Computing*, 34(6):3200–3214, 2022.

A. Ben-Tal, A. Nemirovski, and C. Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

S. Burer and K. Anstreicher. Second-order-cone constraints for extended trust-region subproblems. *SIAM Journal on Optimization*, 23(1):432–451, 2013. doi: 10.1137/110826862. URL https://doi.org/10.1137/110826862.

M. Bynum, G. Hackebeil, W. Hart, C. Laird, B. Nicholson, J. Siirola, J.-P. Watson, and D. Woodruff. *Pyomo–Optimization Modeling in Python*, volume 67. Springer, third edition, 2021.

J. Chen and S. Burer. Globally solving nonconvex quadratic programming problems via completely positive programming. *Mathematical Programming Computation*, 4(1):33–52, 2012.

G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.

M. Dür. Dual bounding procedures lead to convergent branch-and-bound algorithms. *Mathematical Programming*, 91(1):117–125, 2001. doi: 10.1007/s101070100236. URL https://doi.org/10.1007/s101070100236.

C. Eijgenraam, R. Brekelmans, D. den Hertog, and C. Roos. Optimal strategies for flood prevention. *Management Science*, 63(5):1644–1656, 2017.

J. Falk and R. Soland. An algorithm for separable nonconvex programming problems. *Management Science*, 15(9):550–569, 1969.

B. Gorissen, A. Ben-Tal, H. Blanc, and D. den Hertog. Deriving robust and globalized robust solutions of uncertain linear programs with general convex uncertainty sets. *Operations Research*, 62(3):672–679, 2014.

I. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, 26:83–100, 2003.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026. URL https://www.gurobi.com. Gurobi v13.0.0 used in experiments.

W. Hart, J.-P. Watson, and D. Woodruff. Pyomo: Modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260, 2011.

P. Hartman. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3):707–713, 1959.

H. Hof and M. Walter. Geometry of the reformulation-linearization-technique: Domination of disjunctions. *CoRR*, abs/2511.13805, 2025. URL http://arxiv.org/abs/2511.13805.

C. Hojny, M. Besançon, K. Bestuzheva, S. Borst, A. Chmiela, J. Dionísio, L. Eifler, M. Ghannam, A. Gleixner, A. Göß, A. Hoen, R. van der Hulst, D. Kamp, T. Koch, K. Kofler, J. Lentz, S. Maher, G. Mexi, E. Mühmer, M. E. Pfetsch, S. Pokutta, F. Serrano, Y. Shinano, M. Turner, S. Vigerske, M. Walter, D. Weninger, and L. Xu. The SCIP Optimization Suite 10.0. Technical report, Optimization Online, November 2025. URL https://optimization-online.org/2025/11/the-scip-optimization-suite-10-0/.

R. Horst. An algorithm for nonconvex programming problems. *Mathematical Programming*, 10:312–321, 1976.

ILOG, Inc. *ILOG CPLEX 12.7, User Manual*, 2017. URL https://www.ibm.com/support/knowledgecenter/SSSA5P_12.7.0/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html.

R. Jiang and D. Li. Second order cone constrained convex relaxations for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 75:461–494, 2019.

R. Jiang and D. Li. Semidefinite programming based convex relaxation for nonconvex quadratically constrained quadratic programming. In Hoai An Le Thi, Hoai Minh Le, and Tao Pham Dinh, editors, *Optimization of Complex Systems: Theory, Models, Algorithms and Applications*, pages 213–220, Cham, 2020. Springer International Publishing. ISBN 978-3-030-21803-4.

JuliaLang contributors. Julia programming language, v1.12.5. https://julialang.org/downloads/manual-downloads/, 2026. Accessed: 2026-02-14.

JuMP developers. Jump.jl v1.29.4 release notes. https://jump.dev/JuMP.jl/stable/release_notes/, 2026. Version 1.29.4 (January 16, 2026); accessed 2026-02-14.

R. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.

R. Rockafellar and R. Wets. *Variational Analysis*. Springer, 2009.

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

Nikolaos V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, 1996. doi: 10.1007/BF00138693.

A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming*, 124(1):383–411, 2010.

S. Schaible. Parameter-free convex equivalent and dual programs of fractional programming problems. *Zeitschrift f ur Operations Research*, 18(5):187–196, 1974.

A. Selvi, A. Ben-Tal, R. Brekelmans, and D. den Hertog. Convex maximization via adjustable robust optimization. *INFORMS Journal on Computing*, 34(4):2091–2105, 2022.

H. Sherali and W. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

H. Sherali and W. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, volume 31. Springer Science & Business Media, 1999.

H. Sherali and W. Adams. A hierarchy of relaxations leading to the convex hull representation for general discrete optimization problems. *Annals of Operations Research*, 140:21–47, 2005.

H. Sherali and W. Adams. A reformulation-linearization technique for semi-infinite and convex programs under mixed 0-1 and general discrete restrictions. *Discrete Applied Mathematics*, 157(6):1319–1333, 2009.

H. Sherali and A. Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization*, 2(4):379–410, 1992.

H. Sherali and C. Tuncbilek. A global optimization algorithm for polynomial programming problems using a Reformulation–Linearization Technique. *Journal of Global Optimization*, 2(1):101–112, 1992.

J. Sturm and S. Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.

P. Tao and L. An. Convex analysis approach to d. c. programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

The Optimization Firm, LLC. BARON version 21.1.13. Computer software, 2021. URL https://minlp.com/older-baron-versions. Release dated January 13, 2021.

A. Waechter, C. Laird, F. Margot, and Y. Kawajir. Introduction to ipopt: A tutorial for downloading, installing, and using ipopt. *Revision*, 2009. URL https://coin-or.github.io/Ipopt/.

B. Yang and S. Burer. A two-variable approach to the two-trust-region subproblem. *SIAM Journal on Optimization*, 26(1):661–680, 2016.

J. Zhen, A. Marandi, D. de Moor, D. den Hertog, and L. Vandenberghe. Disjoint bilinear optimization: A two-stage robust optimization perspective. *INFORMS Journal on Computing*, 34(5):2410–2427, 2022.

J. Zhen, D. Kuhn, and W. Wiesemann. A unified theory of robust and distributionally robust optimization via the primal-worst-equals-dual-best principle. *Operations Research*, 2023. URL https://pubsonline.informs.org/doi/10.1287/opre.2021.0268.

Appendix

A. Branch and bound strategies comparison

We compare three branching strategies for RPT-BB (*None*, *Secant*, and *SOC*) and two branching strategies for RPT-SDP-BB (*None* and *Secant*) across the five numerical experiments described in Section 8. Under the *None* strategy, branching only partitions the feasible region using the hyperplane

$$\boldsymbol{\nu}^\top \boldsymbol{x} = l_j,$$

where $\boldsymbol{\nu}$ is the eigenvector associated with the largest absolute eigenvalue of $\boldsymbol{X}^* - \boldsymbol{x}^*(\boldsymbol{x}^*)^\top$, and $l_j = \boldsymbol{\nu}^\top \boldsymbol{x}^*$. No additional convex inequalities are added, so branching only splits the current node relaxation into child relaxations. Under the *Secant* strategy, convex secant inequalities are introduced after branching to approximate the nonconvex constraint

$$\boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu} \leq (\boldsymbol{\nu}^\top \boldsymbol{x})^2$$

whenever the corresponding eigenvalue is positive. Under the *SOC* strategy, additional second-order cone (SOC) constraints are imposed whenever the eigenvalue is negative, thereby strengthening the relaxation through the convex inequality

$$(\boldsymbol{\nu}^\top \boldsymbol{x})^2 \leq \boldsymbol{\nu}^\top \boldsymbol{X} \boldsymbol{\nu}.$$

The SOC strategy is not applicable to RPT-SDP-BB, because in that case the matrix $\boldsymbol{X}^* - \boldsymbol{x}^*(\boldsymbol{x}^*)^\top$ is positive semidefinite and therefore cannot have negative eigenvalues.

Results are reported in Tables 9–13. We assess performance using two criteria: (a) computation time and (b) the number of runs that hit the time limit of 600 seconds, reported in the *TO* column. Equivalently, the latter also indicates how many runs were solved within the time limit. Within each approach and for each instance, the fastest runtime is identified, and all runtimes within 5% of that best value are highlighted in bold. This tolerance accounts for minor numerical and computational variability, so runtimes within 5% are regarded as practically equivalent.

The experiments were run on the same computational environment as in Section 8, namely a MacBook Pro with an Apple M3 Pro chip (11 CPU cores: 5 performance and 6 efficiency) and 18 GB RAM, using the same software stack described there.

A.1. Sum-of-max-linear-terms maximization over convex and nonconvex constraints

The results for this comparison are given in Table 9.

RPT-BB. For $\mathcal{X}_1$, all three branching strategies perform similarly. All instances are solved to optimality without timeouts, and runtimes are nearly identical across strategies for most cases.

Sum-of-max-linear-terms maximization over convex and nonconvex constraints

RPT-BB

$\mathcal{X}$	#	None				Secant					SOC					
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO	Opt	Time	Hyp	SOC Hyp	Sec Hyp	TO
$\mathcal{X}_1$	1	23.29	**0.01**	0.0	0/1	23.29	**0.01**	0.0	0.0	0/1	23.29	**0.01**	0.0	0.0	0.0	0/1
	1r	22.72	**0.01**	0.0	0/10	22.72	**0.01**	0.0	0.0	0/10	22.72	**0.01**	0.0	0.0	0.0	0/10
	2	233.94	**0.07**	0.0	0/1	233.94	0.09	0.0	0.0	0/1	233.94	0.10	0.0	0.0	0.0	0/1
	2r	211.87	**0.14**	0.2	0/10	211.87	0.19	0.2	0.0	0/10	211.87	0.23	0.2	0.2	0.0	0/10
	3	1081.62	**104.37**	74.0	0/1	1081.62	115.15	73.0	48.0	0/1	1081.62	117.64	115.0	16.0	99.0	0/1
	3r	1159.90	69.77	41.7	0/10	1159.90	**51.95**	37.0	25.6	0/10	1159.90	55.85	53.4	4.7	48.7	0/10
	4	113.70	**0.02**	0.0	0/1	113.70	**0.02**	0.0	0.0	0/1	113.70	**0.02**	0.0	0.0	0.0	0/1
	4r	83.78	0.08	0.4	0/10	83.78	0.09	0.4	0.4	0/10	83.78	**0.07**	0.5	0.1	0.4	0/10
	5	3002.44	1.81	1.0	0/1	3002.44	1.81	1.0	1.0	0/1	3002.44	**1.47**	1.0	0.0	1.0	0/1
	5r	2898.05	11.83	6.8	0/10	2898.05	**11.00**	6.5	5.1	0/10	2898.05	54.74	6.6	4.0	2.6	0/10
$\mathcal{X}_2$	1	14.58	**0.02**	0.0	0/1	14.58	**0.02**	0.0	0.0	0/1	14.58	**0.02**	0.0	0.0	0.0	0/1
	1r	14.54	0.03	0.0	0/10	14.54	**0.02**	0.0	0.0	0/10	14.54	**0.02**	0.0	0.0	0.0	0/10
	2	136.22	0.45	0.0	0/1	136.22	0.38	0.0	0.0	0/1	136.22	**0.28**	0.0	0.0	0.0	0/1
	2r	122.21	2.05	2.5	0/10	122.21	2.68	3.7	2.1	0/10	122.21	**1.58**	3.1	1.0	2.1	0/10
	3	837.94	600.00*	161.0	1/1	837.94	**325.78**	104.0	79.0	**0/1**	837.94	346.64	106.0	11.0	95.0	**0/1**
	3r	890.07	506.64**	139.9	8/10	890.07	408.17**	67.5	48.2	**6/10**	890.07	**384.59****	99.8	12.0	87.8	**6/10**
	4	33.73	**1.52**	7.0	0/1	33.73	4.12	16.0	12.0	0/1	33.73	2.89	16.0	2.0	14.0	0/1
	4r	31.81	**0.92**	2.7	0/10	31.81	1.43	5.8	4.5	0/10	31.81	1.41	6.0	1.4	4.6	0/10
	5	1610.69	**600.00***	159.0	1/1	1610.69	**600.00***	1.0	0.0	1/1	1610.69	**600.00***	10.0	10.0	0.0	1/1
	5r	1670.92	**600.00***	150.3	10/10	1670.92	**600.00***	22.4	6.7	10/10	1670.92	**600.00***	95.8	71.6	24.2	10/10
$\mathcal{X}_3$	1	13.45	**63.04**	396.0	0/1	13.45	107.81	676.0	636.0	0/1	13.45	109.11	676.0	0.0	636.0	0/1
	1r	15.01	223.38**	537.0	**2/10**	15.01	**204.52****	388.0	318.1	3/10	15.01	252.56**	307.0	0.3	251.1	4/10
	2	140.86	**58.58**	138.0	0/1	140.88	59.61	138.0	137.0	0/1	140.88	**56.75**	138.0	0.0	137.0	0/1
	2r	129.29	**74.08**	116.0	0/10	129.29	95.89	213.0	208.1	0/10	129.29	101.29	230.6	0.1	225.7	0/10
	3	769.11	376.17	73.0	0/1	768.96	**264.37**	48.0	45.0	0/1	768.96	**261.37**	47.0	3.0	44.0	0/1
	3r	805.98	**317.12****	33.6	**2/10**	805.96	342.95**	35.1	34.9	3/10	805.96	396.11**	31.5	0.2	31.0	4/10
	4	45.34	81.43	151.0	0/1	45.34	**45.08**	84.0	83.0	0/1	45.34	**44.18**	88.0	1.0	87.0	0/1
	4r	44.80	518.96**	409.8	7/10	44.78	**450.05****	351.2	346.3	**6/10**	44.54	491.12**	353.8	2.2	350.1	7/10
	5	2257.31	**600.00***	67.0	1/1	2267.88	**600.00***	91.0	19.0	1/1	2268.17	**600.00***	87.0	58.0	29.0	1/1
	5r	2354.84	**600.00***	81.6	10/10	2359.75	**600.00***	82.0	17.0	10/10	2359.96	**600.00***	72.8	46.0	26.8	10/10

RPT-SDP-BB

$\mathcal{X}$	#	None				Secant				
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO
$\mathcal{X}_1$	1	23.29	**0.02**	0.0	0/1	23.29	**0.02**	0.0	0.0	0/1
	1r	22.72	**0.02**	0.0	0/10	22.72	**0.02**	0.0	0.0	0/10
	2	233.94	**0.57**	0.0	0/1	233.94	0.61	0.0	0.0	0/1
	2r	211.87	**0.52**	0.0	0/10	211.87	**0.51**	0.0	0.0	0/10
	3	1081.62	**30.13**	0.0	0/1	1081.62	32.44	0.0	0.0	0/1
	3r	1159.90	**46.26**	0.3	0/10	1159.90	48.63	0.3	0.3	0/10
	4	113.70	**0.06**	0.0	0/1	113.70	0.09	0.0	0.0	0/1
	4r	83.78	**0.06**	0.0	0/10	83.78	**0.06**	0.0	0.0	0/10
	5	3002.44	80.36	1.0	0/1	3002.44	**71.63**	1.0	1.0	0/1
	5r	2898.05	80.82	1.1	0/10	2898.05	**73.41**	1.1	1.1	0/10
$\mathcal{X}_2$	1	14.58	**0.03**	0.0	0/1	14.56	**0.03**	0.0	0.0	0/1
	1r	14.54	0.04	0.0	0/10	14.54	**0.03**	0.0	0.0	0/10
	2	136.22	1.19	0.0	0/1	136.22	**1.05**	0.0	0.0	0/1
	2r	122.21	2.77	0.7	0/10	122.21	**2.54**	0.7	0.7	0/1
	3	837.94	104.21	0.0	0/1	837.94	**94.39**	0.0	0.0	0/1
	3r	890.07	222.30**	0.6	1/10	890.07	**210.05****	0.6	0.6	1/10
	4	33.73	**1.63**	2.0	0/1	33.73	2.00	2.0	0.0	0/1
	4r	31.81	**0.70**	0.8	0/10	31.81	**0.70**	0.8	0.0	0/10
	5	1610.69	**600.00***	4.0	1/1	1610.69	**600.00***	4.0	4.0	1/1
	5r	1670.92	**600.00***	4.0	10/10	1670.92	**600.00***	4.0	4.0	10/10
$\mathcal{X}_3$	1	13.44	**5.08**	10.0	0/1	13.44	**5.01**	10.0	10.0	0/1
	1r	14.99	20.85	27.7	0/10	14.99	**11.88**	18.7	18.7	0/10
	2	140.62	**26.77**	4.0	0/1	140.62	28.73	4.0	4.0	0/1
	2r	129.08	30.73	4.8	0/10	129.08	**26.74**	3.8	3.8	0/10
	3	768.96	**600.0***	2.0	1/1	768.96	**600.0***	2.0	2.0	1/1
	3r	805.08	**600.0***	2.0	10/10	805.95	**600.0***	2.0	2.0	10/10
	4	45.34	**116.81**	52.0	0/1	45.34	147.34	58.0	58.0	0/1
	4r	44.88	**219.65**	54.0	**1/10**	44.83	261.83	58.1	58.1	3/10
	5	-	**600.0***	2.0	1/1	-	**600.0***	2.0	2.0	1/1
	5r	-	**600.0***	2.0	10/10	-	**600.0***	2.0	2.0	10/10

Table 9 Results for Problem (38) over the feasible regions $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$ for the different branching strategies in RPT-BB and RPT-SDP-BB.

The only notable deviation occurs for instance 5r, where SOC is substantially slower than the other strategies (54.74s, compared with 11.83s for None and 11.00s for Secant). Overall, the three strategies are very close on $\mathcal{X}_1$, with None and Secant appearing slightly more stable than SOC.

For $\mathcal{X}_2$, Secant and SOC perform best overall. In particular, Secant and SOC solve instance 3 to optimality within the time, whereas None hits the time limit on this instance. For the replicated instance 3r, Secant and SOC solve more instances than None to optimality. For the remaining instances, runtime and timeout behavior are broadly similar across strategies, with all three variants reaching the time limit on instances 5 and 5r.

For $\mathcal{X}_3$, no single strategy dominates. None is the fastest strategy for instances 1 and 2r and yields fewer timeouts on the replicated instances 1r and 3r. On the other hand, Secant and SOC are advantageous on instances 3 and 4, where they reduce runtime relative to None. In addition, Secant yields the fewest timeouts for the replicated instance 4r.

RPT-SDP-BB. For $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$, no single branching strategy consistently outperforms the other.

Conclusion. For this benchmark, the results do not reveal a clear winner among the branching strategies, as their relative performance varies across $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$.

A.2. Log-sum-exp maximization over linear constraints

Table 10 indicates that, for RPT-BB, the Secant strategy is marginally the strongest variant overall: it attains the fastest runtime on instances 1–3, ties for best on instance 4, and remains competitive on instance 5. For RPT-SDP-BB, Secant and None perform very similarly: they tie on instance 1, Secant is faster on instances 2 and 3, and None is slightly faster on instances 4 and 5. Overall, Secant has a slight edge, although the differences among the strategies are small.

Log-sum-exp maximization over linear constraints

RPT-BB

$\mathcal{X}$	#	None				Secant					SOC					
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO	Opt	Time	Hyp	SOC Hyp	Sec Hyp	TO
$\mathcal{X}_1$	1	10.01	0.08	0.0	0/10	10.01	**0.06**	0.0	0.0	0/10	10.01	0.09	0.0	0.0	0.0	0/10
	2	40.00	**1.32**	0.0	0/10	40.00	**1.28**	0.0	0.0	0/10	40.00	**1.30**	0.0	0.0	0.0	0/10
	3	6.10	4.09	3.0	0/10	6.10	**1.71**	1.0	0.9	0/10	6.10	**1.73**	1.1	0.1	1.0	0/10
	4	21.96	**0.31**	0.0	0/10	21.96	**0.31**	0.0	0.0	0/10	21.96	0.38	0.0	0.0	0.0	0/10
	5	34.76	**5.44**	0.0	0/10	34.76	**5.47**	0.0	0.0	0/10	34.76	5.86	0.0	0.0	0.0	0/10

RPT-SDP-BB

$\mathcal{X}$	#	None				Secant				
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO
$\mathcal{X}_1$	1	10.01	**0.20**	0.0	0/10	10.01	**0.20**	0.0	0.0	0/10
	2	40.00	34.56	0.0	0/10	40.00	**32.92**	0.0	0.0	0/10
	3	6.10	2.61	0.8	0/10	6.10	**2.20**	0.8	0.8	0/10
	4	21.96	**1.61**	0.0	0/10	21.96	**1.68**	0.0	0.0	0/10
	5	34.76	**124.07**	0.0	0/10	34.76	**124.45**	0.0	0.0	0/10

Table 10 Results for Problem (40) for the different branching strategies in RPT-BB and RPT-SDP-BB.

A.3. Linear multiplicative optimization

Table 11 shows that, for **RPT-BB**, the Secant strategy performs best overall. It consistently achieves the fastest computation times for the first three instances and reduces the number of timeouts relative to the other strategies. For RPT-SDP-BB, None consistently attains the fastest computation times across all instances and records fewer timeouts on instance 5 than Secant (2/10 versus 3/10). Overall, Secant is clearly preferable for RPT-BB, whereas for RPT-SDP-BB the choice between None and Secant depends on whether one prioritizes average runtime or solved-run count on the hardest instance.

Linear multiplicative optimization

RPT-BB

$\mathcal{X}$	#	None				Secant					SOC					
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO	Opt	Time	Hyp	SOC Hyp	Sec Hyp	TO
	1	12.14	1.64	25.7	0/10	12.14	**1.24**	25.6	2.2	0/10	12.14	14.63	105.5	1.3	1.8	0/10
	2	23.47	9.04	51.2	**0/10**	23.47	**6.33**	56.2	18.9	**0/10**	23.47	218.43**	535.5	3.4	35.2	2/10
	3	20.96	194.20**	349.1	3/10	20.96	**26.19**	151.5	79.7	**0/10**	20.96	353.78**	573.7	3.4	22.1	1/10
	4	18.99	**572.45****	1059.6	**8/10**	18.99	**575.91****	1029.5	748.2	**8/10**	18.99	600.00*	268.8	1.4	0.0	10/10
	5	8.90	600.00*	161.1	10/10	8.90	600.00*	149.7	15.1	10/10	8.90	600.00*	92.9	1.5	0.0	10/10

RPT-SDP-BB

$\mathcal{X}$	#	None				Secant				
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO
	1	12.14	**4.70**	20.7	0/10	12.14	6.74	20.7	0.3	0/10
	2	23.47	**18.01**	30.3	0/10	23.47	**18.15**	30.3	5.7	0/10
	3	20.96	**20.49**	25.1	0/10	20.96	**21.14**	25.1	2.2	0/10
	4	18.99	**74.19**	35.7	0/10	18.99	84.61	35.7	15.4	0/10
	5	8.90	**358.74****	34.9	**2/10**	8.90	409.70**	33.3	10.5	3/10

Table 11 Results for Problem (42) for the different branching strategies in RPT-BB and RPT-SDP-BB.

A.4. Quadratically constrained quadratic optimization

From Table 12, we observe that for RPT-BB, the Secant and SOC strategies perform similarly and both clearly outperform the None strategy. In particular, both Secant and SOC solve instance 5 to optimality within the time limit, whereas None fails to do so.

For RPT-SDP-BB, the Secant strategy performs slightly better overall, achieving marginally lower computation times on most instances. However, the differences between Secant and None are small.

Overall, Secant appears to be the best default choice. For RPT-BB, Secant and SOC perform similarly and both outperform None, while for RPT-SDP-BB Secant consistently achieves slightly lower computation times.

A.5. Dike height optimization

Table 13 shows that, for RPT-BB, the picture depends on the time grid. For $\boldsymbol{t}_{\mathrm{ir}}$ and $\boldsymbol{t}_{25}$, all three strategies hit the time limit on all instances, so there is little separation in solved-run count. For $\boldsymbol{t}_{50}$,

Quadratically constrained quadratic optimization

$\mathcal{X}$	#	**RPT-BB**														
		None				Secant					SOC					
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO	Opt	Time	Hyp	SOC Hyp	Sec Hyp	TO
	1	394.75	1.23	1.0	0/1	394.75	1.00	1.0	1.0	0/1	394.75	**0.77**	1.0	0.0	1.0	0/1
	2	884.75	0.08	0.0	0/1	884.75	0.06	0.0	0.0	0/1	884.75	**0.05**	0.0	0.0	0.0	0/1
	3	6888.78	**0.01**	0.0	0/1	6888.78	**0.01**	0.0	0.0	0/1	6888.78	**0.01**	0.0	0.0	0.0	0/1
	4	98382.63	7.13	3.0	0/1	98382.63	2.41	1.0	0.0	0/1	98382.63	**1.42**	1.0	0.0	0.0	0/1
	5	-	600.00*	6.0	1/1	774482.39	**87.03**	1.0	0.0	**0/1**	774482.39	93.63	1.0	0.0	0.0	**0/1**
	6	1717.80	0.10	2.0	0/1	1717.80	**0.04**	1.0	0.0	0/1	1717.80	0.07	1.0	1.0	0.0	0/1
	7	4507.75	0.13	4.0	0/1	4507.75	0.06	2.0	2.0	0/1	4507.75	**0.04**	2.0	0.0	2.0	0/1
	8	17084.82	0.23	3.0	0/1	17084.82	0.17	3.0	2.0	0/1	17084.82	**0.16**	3.0	1.0	2.0	0/1
	9	-	600.00*	1336.0	1/1	-	600.00*	1773.0	1061.0	1/1	-	600.00*	1343.0	604.0	739.0	1/1
	10	-	600.00*	532.0	1/1	-	600.00*	880.0	329.0	1/1	-	600.00*	293.0	156.0	137.0	1/1

$\mathcal{X}$	#	**RPT-SDP-BB**								
		None				Secant				
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO
	1	394.75	1.18	1.0	0/1	394.75	**1.12**	1.0	1.0	0/1
	2	884.75	0.13	0.0	0/1	884.75	**0.10**	0.0	0.0	0/1
	3	6888.78	0.02	0.0	0/1	6888.78	**0.01**	0.0	0.0	0/1
	4	98382.63	2.22	0.0	0/1	98382.63	**2.04**	0.0	0.0	0/1
	5	774482.39	**42.31**	0.0	0/1	774482.39	**43.34**	0.0	0.0	0/1
	6	1717.80	0.03	0.0	0/1	1717.80	**0.02**	0.0	0.0	0/1
	7	4507.75	0.18	2.0	0/1	4507.75	**0.13**	2.0	2.0	0/1
	8	17084.82	0.13	1.0	0/1	17084.82	**0.11**	1.0	1.0	0/1
	9	-	600.00*	299.0	1/1	-	600.00*	526.0	526.0	1/1
	10	52233.41	87.21	46.0	0/1	52233.41	**81.69**	46.0	46.0	0/1

Table 12 Results for the quadratically constrained quadratic optimization problem for the different branching strategies in RPT-BB and RPT-SDP-BB.

however, Secant is clearly the strongest variant: it solves dike ring 10 to optimality within the time limit, whereas both None and SOC hit the time limit, and for dike rings 15 and 16 it is faster than SOC while None again fails to solve the instances within the time limit.

For RPT-SDP-BB, both strategies solve all instances almost instantaneously. Secant is marginally faster on most instances, but the differences are negligible because all computation times are below one second.

Overall, Secant is the most effective choice for this benchmark.

A.6. Overall comparison of branching strategies

Across the five numerical experiments, no single branching strategy consistently dominates in all settings. Nevertheless, the Secant strategy emerges as the most reliable choice overall. For the sum-of-max-linear-terms instances, no clear winner emerges, and relative performance varies across instance classes. For the log-sum-exp benchmark, Secant achieves slightly lower runtimes on average, although the differences relative to None and SOC are small. For the linear multiplicative optimization benchmark, Secant is clearly preferable for RPT-BB, whereas for RPT-SDP-BB None yields slightly lower average runtimes but Secant solves one additional run on the hardest instance. Finally, for both the quadratically constrained quadratic optimization benchmark and the dike height optimization benchmark, Secant is the most natural default choice.

Dike height optimization

RPT-BB

t	#	None				Secant					SOC					
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO	Opt	Time	Hyp	SOC Hyp	Sec Hyp	TO
t_{ir}	10	61.98	600.00*	737.0	1/1	61.98	600.00*	1654.0	430.0	1/1	61.98	600.00*	1520.0	829.0	691.0	1/1
	15	608.74	600.00*	1418.0	1/1	608.74	600.00*	1643.0	389.0	1/1	608.74	600.00*	1389.0	642.0	747.0	1/1
	16	1268.11	600.00*	733.0	1/1	1268.11	600.00*	1839.0	380.0	1/1	1268.11	600.00*	1304.0	738.0	566.0	1/1
t_{25}	10	61.31	600.00*	863.0	1/1	61.31	600.00*	1176.0	412.0	1/1	61.31	600.00*	1112.0	528.0	584.0	1/1
	15	609.92	600.00*	647.0	1/1	609.92	600.00*	1182.0	130.0	1/1	609.92	600.00*	719.0	512.0	207.0	1/1
	16	1269.63	600.00	476.0	1/1	1269.63	600.00*	1192.0	317.0	1/1	1269.63	600.00*	818.0	257.0	307.0	1/1
t_{50}	10	55.50	600.00*	326.0	1/1	55.50	**580.42**	3229.0	1060.0	**0/1**	55.50	600.00*	2734.0	1092.0	1642.0	1/1
	15	545.23	600.00*	600.0	1/1	545.23	**146.77**	1078.0	191.0	**0/1**	545.23	194.95	1266.0	585.0	681.0	**0/1**
	16	1100.07	600.00*	439.0	1/1	1100.07	**71.62**	616.0	380.0	**0/1**	1100.07	102.04	564.0	257.0	307.0	**0/1**

RPT-SDP-BB

t	#	None				Secant				
		Opt	Time	Hyp	TO	Opt	Time	Hyp	Sec Hyp	TO
t_{ir}	10	61.98	0.18	0	0/1	61.98	**0.15**	0	0	0/1
	15	608.74	0.17	0	0/1	608.74	**0.16**	0	0	0/1
	16	1268.11	0.15	0	0/1	1268.11	**0.14**	0	0	0/1
t_{25}	10	61.31	0.69	0	0/1	61.31	**0.68**	0	0	0/1
	15	609.92	**0.22**	0	0/1	609.92	0.23	0	0	0/1
	16	1269.63	0.26	0	0/1	1269.63	**0.24**	0	0	0/1
t_{50}	10	55.50	**0.06**	0	0/1	55.50	**0.06**	0	0	0/1
	15	545.23	**0.07**	0	0/1	545.23	**0.07**	0	0	0/1
	16	1100.07	0.15	0	0/1	1100.07	**0.14**	0	0	0/1

Table 13 Results for Problem (DHO).

B. When infimum is not attained

If the infimum of (3) is not attained, we assume that (1) satisfies the following regularity condition.

ASSUMPTION 3. *There exists a vector* $\boldsymbol{x}^{\mathrm{S}} \in \mathrm{ri}(\cap_{k\in\mathcal{K}_0} \mathrm{dom}(f_k))$ *such that* $f_k(\boldsymbol{x}^{\mathrm{S}}) < 0$ *for all* $k \in \mathcal{K}$, $\boldsymbol{A}^\top \boldsymbol{x}^{\mathrm{S}} < \boldsymbol{b}$ *and* $\boldsymbol{h}(\boldsymbol{x}^{\mathrm{S}}) < \boldsymbol{0}$.

Note that Assumption 3 implies that $\boldsymbol{x}^{\mathrm{S}}$ resides in the sets $\cap_{\substack{k\in\mathcal{K}_0\\ i\in\mathcal{I}_0}} \mathrm{ri}(\mathrm{dom}(c_{ik}))$ and $\cap_{j\in\mathcal{J}_0} \mathrm{ri}(\mathrm{dom}(h_j))$ thanks to Proposition 2.42 in Rockafellar and Wets (2009), and thus, $\boldsymbol{x}^{\mathrm{S}}$ is a strict Slater point of (1). Furthermore, there exists a $(\boldsymbol{\tau}^{\mathrm{S}}, \boldsymbol{X}^{\mathrm{S}}, \boldsymbol{V}^{\mathrm{S}})$ such that $(\boldsymbol{x}^{\mathrm{S}}, \boldsymbol{\tau}^{\mathrm{S}}, \boldsymbol{X}^{\mathrm{S}}, \boldsymbol{V}^{\mathrm{S}})$ is a strict Slater point of the corresponding RPT relaxation (8) of (1) with $f_k(\boldsymbol{x}) \leq 0$ is replaced by

$$\exists \boldsymbol{y} : \begin{cases} c_{0k}(\boldsymbol{x}) - \sup\limits_{y\in\mathrm{dom}(c^*_{1k})} \{\boldsymbol{x}^\top \boldsymbol{y} - c^*_{1k}(\boldsymbol{y})\} \leq 0 \\ \boldsymbol{y} \in \mathrm{dom}(c^*_{1k}). \end{cases} \tag{46}$$

Finally, thanks to Remark 1 and the proof of Theorem 6(iii) of Zhen et al. (2023), the inf operator in the constraint of (8) can be merged with the inf operator (instead of min operator because the optimal $\boldsymbol{y}$ may not be obtained) in the objective function without affecting the infimum of (8).

C. Mountain climbing procedure

We use a mountain climbing (MC) procedure based on the algorithm from Tao and An (1997), to find an upper bound for problems involving the biconjugate. The MC procedure takes as input $\mathcal{X}''$, the list of candidate vectors obtained from the solution of the RPT relaxation (see Section 3.4) and returns a local optimum. The procedure is summarized in Algorithm 3, for the problem of maximizing a function $f(\boldsymbol{x},\boldsymbol{y})$ over $\mathcal{X}\times\mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are disjoint sets. Note that it is possible that $\mathcal{X}''=\emptyset$, in which case MC cannot be applied.

Algorithm 3 Mountain climbing procedure

Input: $\mathcal{X}'$, $\mathcal{L}=\emptyset$.

for $\boldsymbol{x}\in\mathcal{X}'$ **do**
  $\boldsymbol{y}\leftarrow \arg\max_{\boldsymbol{y}\in\mathcal{Y}} f(\boldsymbol{x},\boldsymbol{y})$
  $\varepsilon\leftarrow 1$
  while $\varepsilon > 0.001$ **do**
    $\mathrm{Lb}\leftarrow f(\boldsymbol{x},\boldsymbol{y})$
    $\boldsymbol{x}\leftarrow \arg\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x},\boldsymbol{y})$
    $\boldsymbol{y}\leftarrow \arg\max_{\boldsymbol{y}\in\mathcal{Y}} f(\boldsymbol{x},\boldsymbol{y})$
    $\mathrm{Lb}_{\boldsymbol{x}}\leftarrow f(\boldsymbol{x},\boldsymbol{y})$
    $\varepsilon\leftarrow \mathrm{Lb}_{\boldsymbol{x}}-\mathrm{Lb}$
  end while
  $\mathcal{L}\leftarrow\mathcal{L}\cup\{(\boldsymbol{x},\boldsymbol{y})\}$
end for
$(\boldsymbol{x}^\star,\boldsymbol{y}^\star)\leftarrow \arg\max_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{L}} f(\boldsymbol{x},\boldsymbol{y})$
$\mathrm{Lb}^\star=f(\boldsymbol{x}^\star,\boldsymbol{y}^\star)$
return $(\mathrm{Lb}^\star,\boldsymbol{x}^\star,\boldsymbol{y}^\star)$

D. RPT-SDP formulations of the numerical experiments

Throughout the experiments we consider five cases of the feasible set $\mathcal{X}$: $\mathcal{X}=\mathcal{X}_1$, $\mathcal{X}=\mathcal{X}_2$, $\mathcal{X}=\mathcal{X}_3$, $\mathcal{X}=\mathcal{X}_4$, and $\mathcal{X}=\mathcal{X}_5$, where

$$\begin{aligned}
\mathcal{X}_1 &= \{\boldsymbol{x}\in\mathbb{R}^{n_x}_+ \mid \boldsymbol{D}^\top\boldsymbol{x}\le\boldsymbol{d}\}\\
\mathcal{X}_2 &= \left\{\boldsymbol{x}\in\mathcal{X}_1 \;\middle|\; \log\left(\sum_{i=1}^{n_x}\exp(x_i)\right)\le a\right\}\\
\mathcal{X}_3 &= \left\{\boldsymbol{x}\in\mathcal{X}_1 \;\middle|\; \left\|\boldsymbol{x}\right\|_2+\sum_{i=1}^{n_x}\sqrt{x_i}\le c\right\}\\
\mathcal{X}_4 &= \left\{\boldsymbol{x}\in\mathcal{X}_1 \middle| \boldsymbol{A}_i^\top\boldsymbol{x}+b_i\ge 0, i\in\{1,\dots,n_y\}\right\}\\
\mathcal{X}_5 &= \left\{\boldsymbol{x}\in\mathcal{X}_1 \middle| \boldsymbol{x}^\top\boldsymbol{P}_k\boldsymbol{x}+\boldsymbol{q}_k^\top\boldsymbol{x}+r_k\le 0,\ k\in\mathcal{K}_{\mathrm{C}}\right\}.
\end{aligned}$$

We notice that both $\mathcal{X}_2$ and $\mathcal{X}_3$ are not in conic form, but they can be reformulated as such, in the following way. First, for $\mathcal{X}_2$ we observe that $\log\left(\sum_{i=1}^{n_x}\exp(x_i)\right)\le a \iff \sum_{i=1}^{n_x}\exp(x_i-a)\le 1$. Using epigraphical variables z_i we obtain the following equivalent form:

$$\mathcal{X}_2=\left\{\boldsymbol{x}\in\mathcal{X}_1,\boldsymbol{z}\in\mathbb{R}^{n_x} \middle| z_i\ge\exp\left(x_i-a\right),\sum_{i=1}^{n_x}z_i\le 1\right\}.$$

Regarding $\mathcal{X}_3$ we first reformulate the nonconvex constraint via the biconjugate and obtain the equivalent set

$$\mathcal{X}_3 = \left\{ \boldsymbol{x} \in \mathcal{X}_1 \middle| \left\|\boldsymbol{x}\right\|_2 + \sum_{i=1}^{n_x} \frac{1}{4z_i} + \boldsymbol{x}^\top \boldsymbol{z} \leq 0 \right\}.$$

We introduce epigraphical variables for the convex component of the SLC constraint. Since the convex component of the SLC constraint consists of a sum of two basic cone functions we introduce an epigraphical variable for each basic cone function. Subsequently, we reformulate every convex constraint in terms of one of the basic cone constraints. Next, we convexify the SLC constraint, such that we obtain the following relaxed set of constraints

$$\mathcal{X}_3^* = \left\{ \begin{array}{c} \boldsymbol{x} \in \mathcal{X}_1, \boldsymbol{V} \in \mathbb{R}^{n_x \times n_x}, \boldsymbol{z} \in \mathbb{R}^{n_x}_{++}, \\ \boldsymbol{t} \in \mathbb{R}^{n_x}_{++}, s \in \mathbb{R}, p \in \mathbb{R}_{++} \end{array} \middle| \begin{array}{l} s + p + \sum_{i=1}^{n_x} V_{ii} \leq c \\ \left\|\boldsymbol{x}\right\|_2 \leq s \\ \sum_{i=1}^{n_x} t_i \leq p \\ \left\|(z_i - t_i, 1)^\top\right\|_2 \leq z_i + t_i, \quad i \in \{1, \ldots, n_x\} \end{array} \right\}.$$

We choose c to be large enough such that (38) with $\mathcal{X} = \mathcal{X}_3$ satisfies Assumption 1.

When applying the mountain climbing procedure for $\mathcal{X} = \mathcal{X}_3$, we only apply it for the candidate vectors that are feasible, while for $\mathcal{X} = \mathcal{X}_2$ and $\mathcal{X} = \mathcal{X}_3$, we alternate between maximizing for $\boldsymbol{x}, \boldsymbol{z} \in \mathcal{X}$ and maximizing for $\boldsymbol{y} \in \mathcal{Y}$ and vice versa.

In the formulations for the numerical experiments we encounter several products of variables. These are linearized as follows: We linearize $\boldsymbol{x}\boldsymbol{x}^\top$ by $\boldsymbol{X}$, $\boldsymbol{y}\boldsymbol{y}^\top$ by $\boldsymbol{Y}$, $\boldsymbol{z}\boldsymbol{z}^\top$ by $\boldsymbol{Z}$, $\boldsymbol{w}\boldsymbol{w}^\top$ by $\boldsymbol{W}$, $\boldsymbol{t}\boldsymbol{t}^\top$ by $\boldsymbol{T}$, $\boldsymbol{x}\boldsymbol{y}^\top$ by $\boldsymbol{U}$, $\boldsymbol{x}\boldsymbol{z}^\top$ by $\boldsymbol{V}$, $\boldsymbol{x}\boldsymbol{w}^\top$ by $\boldsymbol{Q}$, $\boldsymbol{x}\boldsymbol{t}^\top$ by $\boldsymbol{F}$, $\boldsymbol{y}\boldsymbol{z}^\top$ by $\boldsymbol{R}$, $\boldsymbol{y}\boldsymbol{w}^\top$ by $\boldsymbol{P}$, $\boldsymbol{y}\boldsymbol{t}^\top$ by $\boldsymbol{G}$, $\boldsymbol{z}\boldsymbol{w}^\top$ by $\boldsymbol{K}$, $\boldsymbol{z}\boldsymbol{t}^\top$ by $\boldsymbol{H}$, $s\boldsymbol{x}$ by $\boldsymbol{\alpha}$, $s\boldsymbol{y}$ by $\boldsymbol{\beta}$, $s\boldsymbol{z}$ by $\boldsymbol{\gamma}$, $s\boldsymbol{t}$ by $\boldsymbol{\phi}$, s^2 by σ, $p\boldsymbol{x}$ by $\boldsymbol{\lambda}$, $p\boldsymbol{y}$ by $\boldsymbol{\mu}$, $p\boldsymbol{z}$ by $\boldsymbol{\nu}$, $p\boldsymbol{t}$ by $\boldsymbol{\psi}$, ps by ρ and p^2 by π.

D.1. RPT-SDP formulation of Problem (37)

Replacing the objective function with the biconjugate function in (37) we obtain the following equivalent maximization problem

$$\max_{\substack{\boldsymbol{x} \in \mathcal{X} \\ \boldsymbol{y} \in \mathcal{Y}}} \quad (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})^\top \boldsymbol{y}, \tag{CM$_B$}$$

where $\mathcal{Y}$ is given by

$$\mathcal{Y} = \left\{ \boldsymbol{y} \in \mathbb{R}^{n_y}_+ \mid \sum_{j \in \mathcal{J}_\ell} y_j = 1, \ell \in \mathcal{L} \right\}.$$

$\boldsymbol{\mathcal{X}=\mathcal{X}_1}$. The RPT-SDP formulation is given by

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{U},\boldsymbol{X},\boldsymbol{Y}} \quad \mathrm{Tr}(\boldsymbol{U}\boldsymbol{A})+\boldsymbol{b}^\top\boldsymbol{y}$$

$$\text{s.t.} \quad \boldsymbol{D}^\top\boldsymbol{X}_i-\boldsymbol{d}x_i\le\boldsymbol{0}, \qquad i\in\{1,\dots,n_x\}, \tag{47a}$$

$$\boldsymbol{D}^\top\boldsymbol{U}_j-\boldsymbol{d}y_j\le\boldsymbol{0}, \qquad j\in\{1,\dots,n_y\}, \tag{47b}$$

$$\boldsymbol{d}\boldsymbol{x}^\top\boldsymbol{D}+\boldsymbol{D}^\top\boldsymbol{x}\boldsymbol{d}^\top\le\boldsymbol{D}^\top\boldsymbol{X}\boldsymbol{D}+\boldsymbol{d}\boldsymbol{d}^\top, \tag{47c}$$

$$\sum_{j\in\mathcal{J}_\ell} y_j=1, \qquad \ell\in\mathcal{L}, \tag{47d}$$

$$\sum_{j\in\mathcal{J}_\ell}\boldsymbol{U}_j-\boldsymbol{x}=\boldsymbol{0}, \qquad \ell\in\mathcal{L}, \tag{47e}$$

$$\sum_{j\in\mathcal{J}_\ell}\boldsymbol{Y}_j-\boldsymbol{y}=\boldsymbol{0}, \qquad \ell\in\mathcal{L}, \tag{47f}$$

$$\boldsymbol{X},\boldsymbol{Y},\boldsymbol{U}\ge\boldsymbol{0}, \tag{47g}$$

$$\begin{pmatrix}\boldsymbol{X} & \boldsymbol{U} & \boldsymbol{x}\\ \boldsymbol{U}^\top & \boldsymbol{Y} & \boldsymbol{y}\\ \boldsymbol{x}^\top & \boldsymbol{y}^\top & 1\end{pmatrix}\succeq\boldsymbol{0}, \tag{47h}$$

where constraints (47a) - (47g) result from pairwise multiplication of the linear constraints and constraint (47h) results from the additional SDP relaxation.

Observe that $\boldsymbol{x}\in\mathcal{X}_1$ is redundant. The constraint $\boldsymbol{D}^\top\boldsymbol{x}\le\boldsymbol{d}$ is redundant by (47e), (47d) and (47b):

$$\boldsymbol{D}^\top\boldsymbol{x}\le\boldsymbol{d} \iff \boldsymbol{D}^\top\sum_{j\in\mathcal{J}_\ell}\boldsymbol{U}_j-\boldsymbol{d}\le\boldsymbol{0} \iff \boldsymbol{D}^\top\sum_{j\in\mathcal{J}_\ell}\boldsymbol{U}_j-\boldsymbol{d}\sum_{j\in\mathcal{J}_\ell}y_j\le\boldsymbol{0} \iff \sum_{j\in\mathcal{J}_\ell}\left(\boldsymbol{D}^\top\boldsymbol{U}_j-\boldsymbol{d}y_j\right)\le\boldsymbol{0}.$$

The nonnegativity constraint $\boldsymbol{x}\ge\boldsymbol{0}$ is redundant by (47e) and (47g). Moreover, the nonnegativity constraint $\boldsymbol{y}\ge\boldsymbol{0}$ is redundant by (47f) and (47g). Hence, these constraints are not included in the above formulation.

In the RPT-SDP formulation we hence obtain $n_x|\mathcal{L}|+n_y|\mathcal{L}|+n_xn_y+L'n_y+n_x(n_x+1)/2+L'(L'+1)/2+n_xL'+n_y(n_y+1)/2$ additional linear constraints, one additional SDP constraint and $n_x(n_x+1)/2+n_y(n_y+1)/2+n_xn_y$ extra variables.

$\boldsymbol{\mathcal{X}=\mathcal{X}_2}$. The RPT-SDP formulation is given by

$$\max_{\substack{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\\ \boldsymbol{U},\boldsymbol{V},\boldsymbol{R}\\ \boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}}} \quad \mathrm{Tr}(\boldsymbol{U}\boldsymbol{A})+\boldsymbol{b}^\top\boldsymbol{y}$$

$$\text{s.t.} \quad (47\text{a})-(47\text{g})$$

$$\sum_{i=1}^{n_x} z_i\le 1, \tag{48a}$$

$$\sum_{i=1}^{n_x}\boldsymbol{V}_i-\boldsymbol{x}\le\boldsymbol{0}, \tag{48b}$$

$$\sum_{i=1}^{n_x} \boldsymbol{R}_i - \boldsymbol{y} \leq \boldsymbol{0}, \tag{48c}$$

$$\boldsymbol{D}^\top \boldsymbol{x} - \boldsymbol{D}^\top \sum_{i=1}^{n_x} \boldsymbol{V}_i \leq \boldsymbol{d}(1 - \sum_{i=1}^{n_x} z_i), \tag{48d}$$

$$\sum_{i,j=1}^{n_x} Z_{ij} - 2\sum_{i=1}^{n_x} z_i + 1 \geq 0, \tag{48e}$$

$$\exp(x_i - a) \leq z_i, \qquad i \in \{1,\dots,n_x\}, \tag{48f}$$

$$x_j \exp\left(\frac{X_{ij} - a x_j}{x_j}\right) \leq V_{ji}, \qquad i,j \in \{1,\dots,n_x\}, \tag{48g}$$

$$y_j \exp\left(\frac{U_{ij} - a y_j}{y_j}\right) \leq R_{ji}, \qquad i \in \{1,\dots,n_x\}, j \in \{1,\dots,n_y\}, \tag{48h}$$

$$(d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{x}) \exp\left(\frac{d_\ell x_i - a d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{X}_i + a \boldsymbol{D}_\ell^\top \boldsymbol{x}}{d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{x}}\right) \leq d_\ell z_i - \boldsymbol{D}_\ell^\top \boldsymbol{V}_i, \qquad i \in \{1,\dots,n_x\}, \ell \in \{1,\dots,L'\}, \tag{48i}$$

$$\left(1 - \sum_{j=1}^{n_x} z_j\right) \exp\left(\frac{x_i - a - \sum_{j=1}^{n_x} V_{ij} + a\sum_{j=1}^{n_x} z_j}{1 - \sum_{j=1}^{n_x} z_j}\right) \leq z_i - \sum_{j=1}^{n_x} Z_{ji}, \qquad i \in \{1,\dots,n_x\}, \tag{48j}$$

$$\exp(x_i + x_j - 2a) \leq Z_{ij}, \qquad i \leq j \in \{1,\dots,n_x\}, \tag{48k}$$

$$z_j \exp\left(\frac{V_{ij} - a z_j}{z_j}\right) \leq Z_{ij}, \qquad i \leq j \in \{1,\dots,n_x\}, \tag{48l}$$

$$\sum_{j \in \mathcal{J}_k} \boldsymbol{R}_j - \boldsymbol{z} = \boldsymbol{0}, \qquad k \in \mathcal{K}, \tag{48m}$$

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{U} & \boldsymbol{V} & \boldsymbol{x} \\ \boldsymbol{U}^\top & \boldsymbol{Y} & \boldsymbol{R} & \boldsymbol{y} \\ \boldsymbol{V}^\top & \boldsymbol{R}^\top & \boldsymbol{Z} & \boldsymbol{z} \\ \boldsymbol{x}^\top & \boldsymbol{y}^\top & \boldsymbol{z}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0}, \tag{48n}$$

where constraints (48b) - (48e) result from pairwise multiplication of the new linear constraint over $\boldsymbol{z}$ with the previous linear constraints (48g) - (48l) result from pairwise multiplication of the exponential constraint with the linear inequalities and itself, constraint (48m) results from pairwise multiplication of the initial linear constraint over $\boldsymbol{y}$ with $\boldsymbol{z}$ and constraint (48n) results from the additional SDP relaxation.

In the RPT-SDP formulation we hence obtain $2n_x|\mathcal{L}| + n_y|\mathcal{L}| + n_x n_y + L'n_y + n_x(n_x+1)/2 + L'(L'+1)/2 + n_x L' + n_y(n_y+1)/2 + n_y + n_x + L' + 1$ additional linear constraints, $n_x n_y + n_x^2 + L'n_x + n_x(n_x+1) + n_x$ additional exponential constraints, one additional SDP constraint and $n_x(n_x+1) + n_y(n_y+1)/2 + 2n_x n_y + n_x^2$ extra variables.

$\boldsymbol{\mathcal{X} = \mathcal{X}_3}$.

The pairwise multiplication of the linear constraints gives us the following constraints:

(47a) – (47g)

$$s + p + \sum_{i=1}^{n_x} V_{ii} \leq c, \tag{49a}$$

$$\sum_{i=1}^{n_x} t_i \leq p, \tag{49b}$$

$$\boldsymbol{D}^\top \boldsymbol{V}_i \leq z_i \boldsymbol{d}, \qquad i \in \{1, \ldots, n_x\}, \tag{49c}$$

$$\boldsymbol{D}^\top \boldsymbol{F}_i \leq t_i \boldsymbol{d}, \qquad i \in \{1, \ldots, n_x\}, \tag{49d}$$

$$\boldsymbol{D}^\top \boldsymbol{\lambda} \leq p \boldsymbol{d}, \tag{49e}$$

$$\sum_{j \in \mathcal{J}_\ell} \boldsymbol{r}_j - \boldsymbol{z} = \boldsymbol{0}, \qquad \ell \in \mathcal{L}, \tag{49f}$$

$$\sum_{j \in \mathcal{J}_\ell} \boldsymbol{g}_j - \boldsymbol{t} = \boldsymbol{0}, \qquad \ell \in \mathcal{L}, \tag{49g}$$

$$\sum_{j \in \mathcal{J}_\ell} \gamma_j - s = 0, \qquad \ell \in \mathcal{L}, \tag{49h}$$

$$\sum_{j \in \mathcal{J}_\ell} \mu_j - p = 0, \qquad \ell \in \mathcal{L}, \tag{49i}$$

$$\sum_{i=1}^{n_x} \boldsymbol{F}_i \leq \boldsymbol{\lambda}, \tag{49j}$$

$$\sum_{i=1}^{n_x} \boldsymbol{G}_i \leq \boldsymbol{\mu}, \tag{49k}$$

$$\sum_{i=1}^{n_x} \boldsymbol{H}_i \leq \boldsymbol{\nu}, \tag{49l}$$

$$\sum_{i=1}^{n_x} \boldsymbol{T}_i \leq \boldsymbol{\psi}, \tag{49m}$$

$$\sum_{i=1}^{n_x} \psi_i \leq \pi, \tag{49n}$$

$$\sum_{i=1}^{n_x} d_\ell t_i - \boldsymbol{D}_\ell^\top \boldsymbol{F}_i \leq p d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{\lambda}, \qquad \ell \in \{1, \ldots, L'\}, \tag{49o}$$

$$\pi - 2\sum_{i=1}^{n_x} \psi_i + \sum_{i,j=1}^{n_x} T_{ij} \geq 0, \tag{49p}$$

$$\boldsymbol{V}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{G}, \boldsymbol{H}, \boldsymbol{Z}, \boldsymbol{T} \geq \boldsymbol{0}, \tag{49q}$$

$$\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\psi}, \sigma, \pi \geq \boldsymbol{0}. \tag{49r}$$

Further, the pairwise multiplications of the non-linear ones result in the following constraints:

$$\|\boldsymbol{x}\|_2 \leq s, \tag{50a}$$

$$\left\|(z_i - t_i, 1)^\top\right\|_2 \leq z_i + t_i, \qquad i \in \{1, \ldots, n_x\}, \tag{50b}$$

$$\|\boldsymbol{X}_i\|_2 \leq \alpha_i, \qquad i \in \{1, \ldots, n_x\}, \tag{50c}$$

$$\|\boldsymbol{U}_i\|_2 \leq \beta_i, \qquad \imath \in \{1, \ldots, n_y\}, \tag{50d}$$

$$\|\boldsymbol{V}_i\|_2 \leq \gamma_i, \qquad i \in \{1, \ldots, n_x\}, \tag{50e}$$

$$\|\boldsymbol{F}_i\|_2 \leq \phi_i, \qquad i \in \{1, \ldots, n_x\}, \tag{50f}$$

$$\|\boldsymbol{\lambda}\|_2 \leq \rho, \tag{50g}$$

$$\|d_\ell \boldsymbol{x} - \boldsymbol{X}\boldsymbol{D}_\ell\|_2 \leq s d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{\alpha}, \qquad \ell \in \{1, \ldots, L'\}, \tag{50h}$$

$$\left\|\boldsymbol{\lambda} - \sum_{i=1}^{n_x} \boldsymbol{F}_i\right\|_2 \leq \rho - \sum_{i=1}^{n_x} \phi_i, \tag{50i}$$

$$\left\|\boldsymbol{X}\right\|_F \leq \sigma, \tag{50j}$$

$$\left\|(V_{ji} - F_{ji}, x_j)\right\|_2 \leq V_{ji} + F_{ji}, \qquad i,j \in \{1,\dots,n_x\}, \tag{50k}$$

$$\left\|(R_{ji} - G_{ji}, y_j)\right\|_2 \leq R_{ji} + G_{ji}, \qquad i \in \{1,\dots,n_x\}, j \in \{1,\dots,n_y\}, \tag{50l}$$

$$\left\|(Z_{ji} - H_{ji}, z_j)\right\|_2 \leq Z_{ji} + H_{ji}, \qquad i,j \in \{1,\dots,n_x\}, \tag{50m}$$

$$\left\|(H_{ij} - T_{ij}, t_j)\right\|_2 \leq H_{ij} + T_{ij}, \qquad i,j \in \{1,\dots,n_x\}, \tag{50n}$$

$$\left\|(\nu_i - \psi_i, p)\right\|_2 \leq \nu_i + \psi_i, \qquad i \in \{1,\dots,n_x\}, \tag{50o}$$

$$\left\|\left(d_\ell(z_i - t_i) + \boldsymbol{D}_\ell^\top(\boldsymbol{F}_i - \boldsymbol{V}_i), d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{x}\right)\right\|_2 \leq d_\ell(z_i + t_i) - \boldsymbol{D}_\ell^\top(\boldsymbol{F}_i + \boldsymbol{V}_i), \qquad \ell \in \{1,\dots,L'\}, \tag{50p}$$

$$\left\|(\boldsymbol{V}_i - \boldsymbol{F}_i, \boldsymbol{x})\right\|_2 \leq \gamma_i + \phi_i, \qquad i \in \{1,\dots,n_x\}, \tag{50q}$$

$$\left\|\left(\nu_i - \psi_i - \sum_{j=1}^{n_x} H_{ij} + \sum_{j=1}^{n_x} T_{ij}, p - \sum_{j=1}^{n_x} t_j\right)\right\|_2 \leq \nu_i + \psi_i - \sum_{j=1}^{n_x} H_{ij} - \sum_{j=1}^{n_x} T_{ij}, \qquad i \in \{1,\dots,n_x\}, \tag{50r}$$

$$\left\|\begin{pmatrix} Z_{ij} - H_{ij} - H_{ji} + T_{ij} & z_i - t_i \\ z_j - t_j & 1 \end{pmatrix}\right\|_2 \leq Z_{ij} + H_{ij} + H_{ji} + T_{ij}, \qquad i,j \in \{1,\dots,n_x\}, \tag{50s}$$

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{U} & \boldsymbol{V} & \boldsymbol{F} & \boldsymbol{\alpha} & \boldsymbol{\lambda} & \boldsymbol{x} \\ \boldsymbol{U}^\top & \boldsymbol{Y} & \boldsymbol{R} & \boldsymbol{G} & \boldsymbol{\beta} & \boldsymbol{\mu} & \boldsymbol{y} \\ \boldsymbol{V}^\top & \boldsymbol{R}^\top & \boldsymbol{Z} & \boldsymbol{H} & \boldsymbol{\gamma} & \boldsymbol{\nu} & \boldsymbol{z} \\ \boldsymbol{F}^\top & \boldsymbol{G}^\top & \boldsymbol{H}^\top & \boldsymbol{T} & \boldsymbol{\phi} & \boldsymbol{\psi} & \boldsymbol{t} \\ \boldsymbol{\alpha}^\top & \boldsymbol{\beta}^\top & \boldsymbol{\gamma}^\top & \boldsymbol{\phi}^\top & \sigma & \rho & s \\ \boldsymbol{\lambda}^\top & \boldsymbol{\mu}^\top & \boldsymbol{\nu}^\top & \boldsymbol{\psi}^\top & \rho & \pi & p \\ \boldsymbol{x}^\top & \boldsymbol{y}^\top & \boldsymbol{z}^\top & \boldsymbol{t}^\top & s & p & 1 \end{pmatrix} \succeq \boldsymbol{0}. \tag{50t}$$

In the RPT-SDP formulation we hence obtain $(3|\mathcal{L}| + 2L + 7)n_x + (|\mathcal{L}| + 1)n_y + 3n_x n_y + L'n_y + 3n_x(n_x+1)/2 + 3n_x^2 + L'(L'+1)/2 + n_x L' + n_y(n_y+1)/2 + 2|\mathcal{L}| + 2L' + 6$ additional linear constraints, $5n_x^2 + n_x n_y + 5n_x + n_y + 2L' + 3$ additional second order cone constraints, one additional SDP constraint, and $3n_x(n_x+1)/2 + 3n_x^2 + n_y(n_y+1)/2 + 3n_x n_y + 6n_x + 2n_y + 3$ extra variables.

D.2. RPT-SDP formulation of Problem (40)

Replacing the objective function with the biconjugate function in (40) we obtain the following equivalent maximization problem

$$\max_{\substack{\boldsymbol{x} \in \mathcal{X}_1 \\ (\boldsymbol{y},\boldsymbol{w}) \in \mathcal{Y}}} \quad \boldsymbol{x}^\top \boldsymbol{y} + \sum_{i=1}^{n_x} w_i, \tag{LSEM$_B$}$$

where $\mathcal{Y}$ is given by

$$\mathcal{Y} = \left\{ \boldsymbol{y} \in \mathbb{R}_+^{n_x}, \boldsymbol{w} \in \mathbb{R}^{n_x} \;\middle|\; y_i \exp\left(\frac{w_i}{y_i}\right) \leq 1,\, i \in \{1,\dots,n_x\},\, \sum_{i=1}^{n_x} y_i = 1 \right\}.$$

The RPT-SDP formulation is given by

$$\max_{\substack{\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}, \\ \boldsymbol{X},\boldsymbol{Y},\boldsymbol{W}, \\ \boldsymbol{U},\boldsymbol{Q},\boldsymbol{P}}} \quad \mathrm{Tr}(\boldsymbol{U}) + \sum_{i=1}^{n_x} w_i$$

$$
\begin{aligned}
\text{s.t.}\quad & \boldsymbol{x} \in \mathcal{X}_1, && && \text{(51a)}\\
& (\boldsymbol{y}, \boldsymbol{w}) \in \mathcal{Y}, && && \text{(51b)}\\
& \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U} \geq \boldsymbol{0}, && && \text{(51c)}\\
& \boldsymbol{D}^\top \boldsymbol{X}_i - \boldsymbol{d} x_i \leq \boldsymbol{0}, && i \in \{1, \dots, n_x\}, && \text{(51d)}\\
& \boldsymbol{D}^\top \boldsymbol{U}_i - \boldsymbol{d} y_i \leq \boldsymbol{0}, && i \in \{1, \dots, n_x\}, && \text{(51e)}\\
& \boldsymbol{d}\boldsymbol{x}^\top \boldsymbol{D} + \boldsymbol{D}^\top \boldsymbol{x}\boldsymbol{d}^\top \leq \boldsymbol{D}^\top \boldsymbol{X}\boldsymbol{D} + \boldsymbol{d}\boldsymbol{d}^\top, && && \text{(51f)}\\
& \sum_{i \in \{1,\dots,n_x\}} \boldsymbol{U}_i = \boldsymbol{x}, && && \text{(51g)}\\
& \sum_{i \in \{1,\dots,n_x\}} \boldsymbol{Y}_i = \boldsymbol{y}, && && \text{(51h)}\\
& \sum_{i \in \{1,\dots,n_x\}} (\boldsymbol{P})_i^\top = \boldsymbol{w}, && && \text{(51i)}\\
& U_{ji} \exp\left(\frac{Q_{ji}}{U_{ji}}\right) \leq x_j, && i, j \in \{1, \dots, n_x\}, && \text{(51j)}\\
& Y_{ij} \exp\left(\frac{P_{ji}}{Y_{ij}}\right) \leq y_j, && i, j \in \{1, \dots, n_x\}, && \text{(51k)}\\
& \left(d_\ell y_i - \boldsymbol{D}_\ell^\top \boldsymbol{U}_i\right) \exp\left(\frac{d_\ell w_i - \boldsymbol{D}_\ell^\top \boldsymbol{V}_i}{d_\ell y_i - \boldsymbol{D}_\ell^\top \boldsymbol{U}_i}\right) \leq d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{x}, && i \in \{1, \dots, n_x\},\ \ell \in \{1, \dots, L'\}, && \text{(51l)}\\
& Y_{ij} \exp\left(\frac{P_{ji} + P_{ij}}{Y_{ij}}\right) \leq 1, && i \leq j \in \{1, \dots, n_x\}, && \text{(51m)}\\
& \begin{pmatrix} \boldsymbol{X} & \boldsymbol{U} & \boldsymbol{Q} & \boldsymbol{x} \\ \boldsymbol{U}^\top & \boldsymbol{Y} & \boldsymbol{P} & \boldsymbol{y} \\ \boldsymbol{Q}^\top & \boldsymbol{P}^\top & \boldsymbol{W} & \boldsymbol{w} \\ \boldsymbol{x}^\top & \boldsymbol{y}^\top & \boldsymbol{w}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0}, && && \text{(51n)}
\end{aligned}
$$

where constraints (51c) - (51i) result from pairwise multiplication of the linear constraints. Note that we only multiply the linear equality constraint with the variables (see Theorem 1). Constraints (51j) - (51l) result from pairwise multiplication of the linear inequality constraints with the exponential cone constraints, constraint (51m) results from pairwise multiplication of the exponential cone constraints with each other, and constraint (51n) results from the additional SDP relaxation.

In the RPT-SDP formulation we hence obtain $n_x(n_x+1) + n_x^2 + (2L'+3)n_x + L'(L'+1)/2$ additional linear constraints, $2n_x^2 + L'n_x + n_x(n_x+1)/2$ additional exponential cone constraints, one additional SDP constraint and $3n_x(n_x+1)/2 + 3n_x^2$ additional variables.

Observe that we could exclude the nonnegativity constraints from the above reformulation, since from constraints (51b), (51g), and (51h) it follows that they are redundant.

D.3. RPT-SDP formulation of Problem (42)

Replacing the objective function with the biconjugate function in (42) we obtain the following equivalent maximization problem

$$
\max_{\substack{\boldsymbol{x} \in \mathcal{X} \\ (\boldsymbol{y}, \boldsymbol{w}) \in \mathcal{Y}}} \quad -(\boldsymbol{A}^\top \boldsymbol{x} + \boldsymbol{b})^\top \boldsymbol{y} - \sum_{i \in \{1,\dots,n_y\}} w_i, \tag{CM$_B$}
$$

where $\mathcal{Y}$ is given by

$$\mathcal{Y} = \left\{ \boldsymbol{y} \in \mathbb{R}^{n_y}_+, \boldsymbol{w} \in \mathbb{R}^{n_y} \;\middle|\; \exp(-w_i - 1) \le y_i,\, i \in \{1,\ldots,n_y\} \right\}.$$

The RPT-SDP formulation is given by

$$\begin{aligned}
\max_{\substack{\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},\\ \boldsymbol{X},\boldsymbol{Y},\boldsymbol{W},\\ \boldsymbol{U},\boldsymbol{Q},\boldsymbol{P}}} \quad & \operatorname{Tr}(\boldsymbol{U}\boldsymbol{A}) + \boldsymbol{b}^\top y + \sum_{i=1}^{n_y} w_i \\
\text{s.t.} \quad & \boldsymbol{D}^\top \boldsymbol{x} \le \boldsymbol{d}, && (52a)\\
& \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U} \ge \boldsymbol{0}, && (52b)\\
& \boldsymbol{D}^\top \boldsymbol{X}_i - \boldsymbol{d} x_i \le \boldsymbol{0}, \quad i \in \{1,\ldots,n_x\}, && (52c)\\
& \boldsymbol{D}^\top \boldsymbol{U}_j - \boldsymbol{d} y_j \le \boldsymbol{0}, \quad j \in \{1,\ldots,n_y\}, && (52d)\\
& \boldsymbol{d}\boldsymbol{x}^\top \boldsymbol{D} + \boldsymbol{D}^\top \boldsymbol{x}\boldsymbol{d}^\top \le \boldsymbol{D}^\top \boldsymbol{X}\boldsymbol{D} + \boldsymbol{d}\boldsymbol{d}^\top, && (52e)\\
& \exp(-w_i - 1) \le y_i, \quad i \in \{1,\ldots,n_y\}, && (52f)\\
& x_j \exp\left(\frac{-Q_{ji} - x_j}{x_j}\right) \le U_{ji}, \quad i \in \{1,\ldots,n_y\},\, j \in \{1,\ldots,n_x\}, && (52g)\\
& y_j \exp\left(\frac{-P_{ji} - y_j}{y_j}\right) \le Y_{ij}, \quad i,j \in \{1,\ldots,n_y\}, && (52h)\\
& (d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{x}) \exp\left(\frac{\boldsymbol{D}_\ell^\top(\boldsymbol{x} + \boldsymbol{Q}_i) - d_\ell(1 + w_i)}{d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{x}}\right) \\
& \qquad \le d_\ell y_i - \boldsymbol{D}_\ell^\top \boldsymbol{U}_i, \quad i \in \{1,\ldots,n_y\},\, \ell \in \{1,\ldots,L'\}, && (52i)\\
& \exp(-w_i - w_j - 2) \le Y_{ij}, \quad i \le j \in \{1,\ldots,n_y\}, && (52j)\\
& \begin{pmatrix} \boldsymbol{X} & \boldsymbol{U} & \boldsymbol{Q} & \boldsymbol{x} \\ \boldsymbol{U}^\top & \boldsymbol{Y} & \boldsymbol{P} & \boldsymbol{y} \\ \boldsymbol{Q}^\top & \boldsymbol{P}^\top & \boldsymbol{W} & \boldsymbol{w} \\ \boldsymbol{x}^\top & \boldsymbol{y}^\top & \boldsymbol{w}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0}, && (52k)
\end{aligned}$$

where constraints (52b) - (52d) result from pairwise multiplication of the linear constraints, constraints (52g) - (52j) result from pairwise multiplication of the exponential constraints with the linear and constraint (52k) results from the additional SDP relaxation.

In the RPT-SDP formulation we hence obtain $n_x(n_x+1)/2 + n_y(n_y+1)/2 + n_x n_y + L'(n_x+n_y) + L'(L'+1)/2$ additional linear constraints, $n_x n_y + n_y^2 + L' n_y + n_y(n_y+1)/2$ additional exponential cone constraints, one additional SDP constraint and $n_x(n_x+1)/2 + n_y(n_y+1) + 2n_x n_y + n_y^2$ additional variables.

D.4. RPT-SDP formulation of Problem (45)

The convex quadratic constraints ($\mathcal{C}$) are reformulated as second order cone constraints, that is $\left\|\boldsymbol{P}_i^{1/2}\boldsymbol{x}\right\|_2 \leq -r_i - \boldsymbol{q}_i^\top \boldsymbol{x}$. The nonconvex quadratic constraints ($\mathcal{NC}$) are linearized as follows: $tr(\boldsymbol{P}_i\boldsymbol{X}) + q_i^\top \boldsymbol{x} + r_i \leq 0$. The RPT-SDP formulation is given by

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{X}} \quad & \mathrm{Tr}(\boldsymbol{P}_0\boldsymbol{X}) + \boldsymbol{q}_0^\top x + r_0 \\
\text{s.t.} \quad & \boldsymbol{x} \in \mathcal{X}_1, && \text{(53a)}\\
& \boldsymbol{D}^\top \boldsymbol{X}_i - \boldsymbol{d}x_i \leq \boldsymbol{0}, \quad i \in \{1,\dots,n_x\}, && \text{(53b)}\\
& \boldsymbol{d}\boldsymbol{x}^\top \boldsymbol{D} + \boldsymbol{D}^\top \boldsymbol{x}\boldsymbol{d}^\top \leq \boldsymbol{D}^\top \boldsymbol{X}\boldsymbol{D} + \boldsymbol{d}\boldsymbol{d}^\top, && \text{(53c)}\\
& \mathrm{Tr}(\boldsymbol{P}_k\boldsymbol{X}) + \boldsymbol{k}_0^\top x + r_k \leq 0, \quad k \in \mathcal{NC}, && \text{(53d)}\\
& \left\|\boldsymbol{P}_k^{1/2}\boldsymbol{x}\right\|_2 \leq -r_k - \boldsymbol{q}_k^\top \boldsymbol{x}, \quad k \in \mathcal{C}, && \text{(53e)}\\
& \left\|\boldsymbol{P}_i^{1/2}\boldsymbol{X}\boldsymbol{P}_j^{1/2}\right\|_2 \leq r_i r_j + r_i \boldsymbol{q}_j^\top x + r_j \boldsymbol{q}_i^\top \boldsymbol{x} + \boldsymbol{q}_i^\top \boldsymbol{X}\boldsymbol{q}_j, \quad i,j \in \mathcal{C}, && \text{(53f)}\\
& \left\|d_\ell \boldsymbol{P}_k^{1/2}\boldsymbol{x} - \boldsymbol{P}_k^{1/2}\boldsymbol{X}\boldsymbol{D}_\ell\right\|_2 \leq -r_k d_\ell + r_k \boldsymbol{D}_\ell^\top \boldsymbol{x} - d_\ell \boldsymbol{q}_k^\top \boldsymbol{x} + \boldsymbol{q}_k^\top \boldsymbol{X}\boldsymbol{D}_\ell, \quad k \in \mathcal{C},\ \ell \in \{1,\dots,L'\}, && \text{(53g)}\\
& \left\|\boldsymbol{P}_k^{1/2}\boldsymbol{X}_j\right\| \leq -r_k x_j - \boldsymbol{q}_k^\top \boldsymbol{X}_j, \quad k \in \mathcal{C},\ j \in \{1,\dots,n_x\}, && \text{(53h)}\\
& \boldsymbol{X} \geq \boldsymbol{0}, && \text{(53i)}\\
& \begin{pmatrix} \boldsymbol{X} & \boldsymbol{x} \\ \boldsymbol{x}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0}, && \text{(53j)}
\end{aligned}
$$

where constraints (53b) - (53c) and (53i) result from pairwise multiplication of the linear constraints, constraints (53f) - (53h) result from pairwise multiplication of the convex quadratic constraints with the linear constraints and each other and constraint (53j) results from the additional SDP relaxation.

In the RPT-SDP formulation we hence obtain $L'n_x + n_x(n_x+1)/2 + L'(L'+1)/2 + |\mathcal{NC}|$ additional linear constraints, $(n_x + L' + 1)|\mathcal{C}| + |\mathcal{C}|(|\mathcal{C}|+1)/2$ additional second order cone constraints, one additional SDP constraint and $n_x(n_x+1)/2$ additional variables.

D.5. RPT-SDP formulation of Problem (DHO)

We introduce the following epigraphical variables: We use z_k for the nonconvex terms in the objective $(C + bx_k)\exp\left(\lambda \sum_{i=0}^k x_i - \delta t_k\right)$, and w_k for $\exp\left(-\theta \sum_{i=0}^k x_i\right)$. The RPT-SDP formulation is given by

$$
\begin{aligned}
\min_{\substack{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v},\boldsymbol{z},\\ \boldsymbol{q},\boldsymbol{X},\boldsymbol{V},\\ \boldsymbol{S},w,W}} \quad & \sum_{k\in\mathcal{K}} z_k + \sum_{k\in\mathcal{K}_0} \frac{S_0}{\beta_\delta}\left(\exp\left(\beta_\delta t_{k+1}\right) - \exp\left(\beta_\delta t_k\right)\right) w_k + \frac{S_0}{\delta}\exp\left(\beta_\delta T\right) w_K \\
\text{s.t.} \quad & (C + bx_k)\exp\left(\frac{\lambda C h_k + \lambda b \sum_{i=0}^k X_{ik} - \delta t_k (C + bx_k)}{C + bx_k}\right) \leq z_k, \quad k \in \mathcal{K}_0, && \text{(54a)}
\end{aligned}
$$

$$\boldsymbol{D}^\top \boldsymbol{x} \le \boldsymbol{d}, \tag{54b}$$

$$\exp\left(-\theta \sum_{i=0}^{k} x_i\right) \le w_k, \quad k \in \mathcal{K}_0, \tag{54c}$$

$$\boldsymbol{D}^\top \boldsymbol{X}_k \le \boldsymbol{x}_k d, \quad k \in \mathcal{K}_0, \tag{54d}$$

$$\boldsymbol{d}\boldsymbol{x}^\top \boldsymbol{D} + \boldsymbol{D}^\top \boldsymbol{x}\boldsymbol{d}^\top \le \boldsymbol{D}^\top \boldsymbol{X}\boldsymbol{D} + \boldsymbol{d}\boldsymbol{d}^\top, \tag{54e}$$

$$(d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{x}) \exp\left(\frac{-\theta d_\ell \sum_{i=0}^k x_i + \theta \sum_{i=0}^k \boldsymbol{D}_\ell^\top \boldsymbol{X}_i}{d_\ell - \boldsymbol{D}_\ell^\top \boldsymbol{x}}\right) \le d_\ell w_k - \boldsymbol{D}_\ell^\top \boldsymbol{Q}_k, \quad k \in \mathcal{K}_0, \ell \in \{1, \dots, L'\}, \tag{54f}$$

$$x_j \exp\left(\frac{-\theta \sum_{i=0}^k X_{ij}}{x_j}\right) \le Q_{jk}, \quad k, j \in \mathcal{K}_0, \tag{54g}$$

$$\exp\left(-\theta \sum_{i=1}^{k} x_i - \theta \sum_{i=1}^{j} x_i\right) \le W_{jk}, \quad j, k \in \mathcal{K}_0, \tag{54h}$$

$$w_j \exp\left(\frac{-\theta \sum_{i=1}^k Q_{kj}}{w_j}\right) \le W_{jk}, \quad j, k \in \mathcal{K}_0, \tag{54i}$$

$$\boldsymbol{x}, \boldsymbol{X} \ge \boldsymbol{0}, \tag{54j}$$

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{Q} & \boldsymbol{x} \\ \boldsymbol{Q}^\top & \boldsymbol{W} & \boldsymbol{w} \\ \boldsymbol{x}^\top & \boldsymbol{w}^\top & 1 \end{pmatrix} \succeq \boldsymbol{0}, \tag{54k}$$

where constraint (54b) represents the upper bounds on $\boldsymbol{x}$ such that we obtain a compact feasible region, (54d) - (54f) result from pairwise multiplication of (54b) with all other constraints, (54g) results from pairwise multiplication of (54c) with the nonnegativity constraint $\boldsymbol{x} \ge 0$, (54h) - (54i) result from pairwise multiplication of the exponential constraint with itself, (54j) results from pairwise multiplication of the nonnegativity constraints, and constraint (54k) results from the additional SDP relaxation.

In the RPT-SDP formulation we hence obtain $|\mathcal{K}_0| + L'(L'+1)/2 + |\mathcal{K}_0|(|\mathcal{K}_0|+1)/2$ additional linear inequalities, $(L'+1)|\mathcal{K}_0| + 2|\mathcal{K}_0|^2$ additional exponential cone inequalities, and one additional LMI.

E. Data generation of numerical experiments

E.1. Data generation of numerical experiments of Problem (37)

We use the data generated by Selvi et al. (2022, Appendix F.5). Instances 1 - 5 refer to the instances 1, 2, 3, 7, and 11 in Selvi et al. (2022, Appendix F.5) respectively. In every problem, every max-term has the same number of elements, i.e., $|\mathcal{J}_\ell| = |\mathcal{J}_{\ell'}|$ for every $\ell, \ell' \in \mathcal{L}$.

Problem instance 1: $A_{ij} \sim [-5,5], b_j = 0, \boldsymbol{D}^\top = \boldsymbol{I}, d_i = n_x/i,$

Problem instance 2: $A_{ij} \sim [-5,5], b_j = 0, \boldsymbol{D}^\top = \boldsymbol{I}, d_i = n_x/i,$

Problem instance 3: $A_{ij} \sim [-5,5], b_j = 0, \boldsymbol{D}^\top = \boldsymbol{I}, d_i = n_x/i,$

Problem instance 4: $A_{ij} \sim [-5,5], b_j \sim [-10,10], D_{ij} \sim [0,1], d_i \sim [5,15],$

Problem instance 5: $A_{ij} \sim [-5,10], b_j \sim [-10,10],$ and $\boldsymbol{D}$ and $\boldsymbol{d}$ are given by:

$$\boldsymbol{D} = \begin{bmatrix} -3 & 7 & 0 & -5 & 1 & 1 & 0 & 2 & -1 & 1 \\ 7 & 0 & -5 & 1 & 1 & 0 & 2 & -1 & -1 & 1 \\ 0 & -5 & 1 & 1 & 0 & 2 & -1 & -1 & -9 & 1 \\ -5 & 1 & 1 & 0 & 2 & -1 & -1 & -9 & 3 & 1 \\ 1 & 1 & 0 & 2 & -1 & -1 & -9 & 3 & 5 & 1 \\ 1 & 0 & 2 & -1 & -1 & -9 & 3 & 5 & 0 & 1 \\ 0 & 2 & -1 & -1 & -9 & 3 & 5 & 0 & 0 & 1 \\ 2 & -1 & -1 & -9 & 3 & 5 & 0 & 0 & 1 & 1 \\ -1 & -1 & -9 & 3 & 5 & 0 & 0 & 1 & 7 & 1 \\ -1 & -9 & 3 & 5 & 0 & 0 & 1 & 7 & -7 & 1 \\ -9 & 3 & 5 & 0 & 0 & 1 & 7 & -7 & -4 & 1 \\ 3 & 5 & 0 & 0 & 1 & 7 & -7 & -4 & -6 & 1 \\ 5 & 0 & 0 & 1 & 7 & -7 & -4 & -6 & -3 & 1 \\ 0 & 0 & 1 & 7 & -7 & -4 & -6 & -3 & 7 & 1 \\ 0 & 1 & 7 & -7 & -4 & -6 & -3 & 7 & 0 & 1 \\ 1 & 7 & -7 & -4 & -6 & -3 & 7 & 0 & -5 & 1 \\ 7 & -7 & -4 & -6 & -3 & 7 & 0 & -5 & 1 & 1 \\ -7 & -4 & -6 & -3 & 7 & 0 & -5 & 1 & 1 & 1 \\ -4 & -6 & -3 & 7 & 0 & -5 & 1 & 1 & 0 & 1 \\ -6 & -3 & 7 & 0 & -5 & 1 & 1 & 0 & 2 & 1 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{d} = \begin{bmatrix} -5 \\ 2 \\ -1 \\ -3 \\ 5 \\ 4 \\ -1 \\ 0 \\ 9 \\ 40 \end{bmatrix},$$

respectively. The values for the parameters of each distinct problem instance are given in Table 14.

Instance	n_x	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{J}_\ell\rvert$	a	c	M
1	5	1	5	3	6	100
2	5	10	5	3	6	100
3	20	10	10	11	25	1000
4	10	2	5	3	7	1000
5	20	10	10	5	30	1000

Table 14 Problem (38) parameters for each instance. n_x refers to the number of variables, $|\mathcal{L}|$ to the number of max linear terms, $|\mathcal{J}_\ell|$ to the number of elements within a max-term, a to the parameter used in $\mathcal{X}_2$, c to the parameter used in $\mathcal{X}_3$ and M to the big M parameter used in Problem (39).

E.2. Data generation of numerical experiments of Problem (40)

The problem instances are adopted from Selvi et al. (2022) and can be summarized as follows: In instances 1 and 2 the linear constraints are defined as

$$-\frac{i}{n} \leq x_i \leq \frac{i}{n},$$

in instance 3 as

$$x_i \leq 8,\ x_i + x_j \leq u_{ij},$$

where $u_{ij} \sim [5, 15]$. Finally, for the last two we have

$$\textbf{Problem instance 4: } D_{ij} \sim [0,1], d_i \sim [10,30],$$

$$\textbf{Problem instance 5: } D_{ij} \sim [0,1], d_i \sim [20,60].$$

The parameters describing each instance are summarized in Table 15.

Table 15 Problem (40) parameters for each instance. n_x refers to the number of variables and L' to the number of linear constraints.

Instance	n_x	L'
1	10	20
2	40	80
3	10	100
4	20	20
5	50	50

E.3. Data generation of numerical experiments of Problem (42)

The problem instances were generated in the same way as in BARON (Ryoo and Sahinidis, 1996). Namely, the constraint coefficients were generated as $D_{ij} \sim [-100, 0], d_i \sim [-100, 0]$ and the linear terms as $A_{ij} \sim [0, 10], b_i \sim [0, 10]$. The parameters describing each instance are summarized in Table 16.

Table 16 Problem (42) parameters for each instance. n_x refers to the number of variables, L' to the number of linear constraints, and n_y to the number of linear multiplications in the objective.

Instance	n_x	L'	n_y
1	5	5	5
2	7	7	10
3	10	10	9
4	20	20	8
5	40	40	4

E.4. Data generation of numerical experiments of Problem (45)

The first 5 problem instances are adopted from Selvi et al. (2022) and can be summarized as follows: In instances 1 and 2 the objectives are $f(\boldsymbol{x}) = -\frac{1}{2}\sum_{i=1}^{20}(x_i - 2)^2$ and $f(\boldsymbol{x}) = -\frac{1}{2}\sum_{i=1}^{20}(x_i + 5)^2$ respectively and the linear constraints are as in instance 11 for problem (37). Instances 3, 4 and 5 are defined by the linear constraints $\boldsymbol{D}^\top \boldsymbol{x} \leq \boldsymbol{d}, \quad \boldsymbol{x} \leq x_u \boldsymbol{e}$, where

$$\textbf{Problem instance 3: } D_{ij} \sim [0,1], d_i \sim [20,60], x_u = 5,$$

$$\textbf{Problem instance 4: } D_{ij} \sim [0,1], d_i \sim [30,60], x_u = 3,$$

$$\textbf{Problem instance 5: } D_{ij} \sim [0,1], d_i \sim [80,120], x_u = 2.$$

Instances 6, 7, 8, 9 and 10 were adopted from Al-Khayyal et al. (1995). Each matrix $\boldsymbol{P}_i \in \mathbb{R}^{n_x \times n_x}$ in both the objective and the constraints has integer entries uniformly at random between -10 and 10 and further in each row, half of the entries are randomly set to 0. Each vector $\boldsymbol{q}_i \in \mathbb{R}^{n_x}$ is also generated with integer entries between -10 and 10 and each r_i is set to 0. The parameters describing each instance are summarized in Table 17.

Table 17 Problem (45) parameters for each instance. n_x refers to the number of variables, L to the number of linear constraints and nc-q to the number of nonconvex quadratic constraints.

Instance	n_x	L	$nc-q$
1	20	10	0
2	20	10	0
3	10	15	0
4	50	62	0
5	100	130	0
6	8	8	4
7	12	12	6
8	16	16	8
9	30	30	15
10	40	40	20

E.5. Data generation of numerical experiments of Problem (DHO)

The linear constraints are defined as $x_i \leq 300$. Moreover, the time periods in each instance are:

$$\boldsymbol{t}_{25} = (0, 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275)^\top,$$

$$\boldsymbol{t}_{50} = (0, 50, 100, 150, 200, 250)^\top,$$

$$\boldsymbol{t}_{ir} = (0, 20, 50, 90, 130, 155, 180, 210, 255, 270)^\top.$$

In each instance, the number of variables n_x is equal to the number of time periods. The parameters describing each instance are summarized in Table 18. Moreover, we have $\theta = \alpha - \zeta, \ \beta_\delta = \alpha\eta + \gamma - \delta$.

Table 18 Problem (DHO) parameters for each instance.

Instance	α	C	b	λ	ζ	η	S_0	γ	δ	T
10	0.033027	16.6939	0.6258	0.0014	0.003774	0.32	0.68938	0.02	0.04	300
15	0.0502	125.6422	1.1268	0.0098	0.003764	0.76	16.2008	0.02	0.04	300
16	0.0574	324.6287	2.1304	0.01	0.002032	0.76	25.0071	0.02	0.04	300