Tight Convex Relaxations for the Expansion Planning Problem

Ralf Lenz · Felipe Serrano

Received: 13 February 2021 / Accepted: 20 March 2022 / Published online: 22 April 2022
© The Author(s) 2022

Abstract
Secure energy transport is considered as highly relevant for the basic infrastructure of nowadays society and economy. To satisfy increasing demands and to handle more diverse transport situations, operators of energy networks regularly expand the capacity of their network by building new network elements, known as the expansion planning problem. A key constraint function in expansion planning problems is a nonlinear and nonconvex potential loss function. In order to improve the algorithmic performance of state-of-the-art MINLP solvers, this paper presents an algebraic description for the convex envelope of this function. Through a thorough computational study, we show that this tighter relaxation tremendously improves the performance of the MINLP solver SCIP on a large test set of practically relevant instances for the expansion planning problem. In particular, the results show that our achievements lead to an improvement of the solver performance for a development version by up to 58%.

Keywords Convex envelopes · Mixed-integer nonlinear programming · Expansion planning of energy networks

Mathematics Subject Classification 90C30 · 90C26 · 90C90

1 Introduction
Expansion planning problems of energy networks (see Sect. 2) are becoming more relevant as well as more challenging due to increasing demands of modern society.
The goal of this paper is to improve the algorithmic performance of an LP-based branch-and-bound MINLP solver for the expansion planning problem. In particular, we focus on building a tighter LP relaxation by exploiting the convex envelope of the main nonlinearity present in this problem. To this end, we compute the convex envelope and implement it in the MINLP solver SCIP.

The state of the art for solving nonconvex MINLPs to global optimality is spatial branch-and-bound. Within spatial branch-and-bound one of the most important ingredients is the construction of tight convex relaxations. For a nonconvex constraint \( \{ x \in D \mid f(x) \leq 0 \} \), where \( D \subseteq \mathbb{R}^n \) is convex, the most common method for constructing a convex relaxation is to compute a convex underestimator, that is, a function \( g : D \rightarrow \mathbb{R} \) that satisfies \( g(x) \leq f(x) \) for all \( x \in D \). Then, \( g(x) \leq 0 \) yields a convex relaxation of the constraint \( f(x) \leq 0 \).

The best possible convex underestimator of a nonconvex function \( f \) over the domain \( D \) is given by the convex envelope \( \text{vex}_D(f) \). For general functions, computing the convex envelope is a difficult task but for specific functions it might be tractable. Note that constraints can also be of the form \( f(x) \geq 0 \), in which case concave overestimators are required to build a convex relaxation. In particular, for equality constraints, both the convex under and concave overestimators are needed.

In this paper, we analytically derive the convex and concave envelope of the potential loss function \( f(x, y) = y \text{sgn}(x) |x|^{\alpha} \) with \( \alpha > 1 \). This is the only source of nonlinearity and nonconvexity of the expansion planning problems that we consider, see Sect. 2. Thus, the convex relaxations depend on the convexification of these constraints. As we will see in Sect. 4.1, the concave envelope of \( f \) can be deduced from the convex envelope of \( f \), and thus, we concentrate on the convex envelope in the following. To the best of our knowledge, the convex envelope of this function is unknown in the literature.

**Contributions**

(i) We derive an explicit description for the convex and concave envelope of \( f \) over a rectangular domain \( D = [x, \bar{x}] \times [y, \bar{y}] \) with \( x < 0 < \bar{x} \) and \( 0 < y < \bar{y} \).

(ii) We perform an exhaustive computational study, we show that these tighter relaxations tremendously improve the performance of the MINLP solver SCIP on a large test set of practically relevant instances that stem from real-world applications. Our results show that the performance improves up to 58% for hard instances.

**Outline** The rest of the paper is structured as follows: Sect. 2 introduces the expansion planning problem. In Sect. 3, we introduce preliminary theoretical results from convex analysis that we need to derive the convex envelope of \( y \text{sgn}(x) |x|^{\alpha} \). In Sect. 4, we present our main contribution, which is the derivation of a closed-form expression for the convex envelope of \( f \). In Sect. 5, we compare the convex envelope with the standard factorable relaxation for the constraint function \( f \) followed by an extensive computational study in Sect. 6. We test the impact of the convex envelope over real-world expansion planning problem instances using the MINLP solver SCIP. To this end, we first study the impact on the dual bound in the root node and secondly the overall performance within the spatial branch-and-bound search. Finally, Sect. 7 provides conclusions.
2 Expansion Planning Problem

Transmission system operators of energy networks such as natural gas, hydrogen and water networks regularly face both increasing demand and more diverse transport situations. To deal with these challenges, they must expand the capacity of the existing networks without having to resort to the expensive options of setting up new pipeline corridors or replacing pipelines by larger ones. In practice, operators preferably build new pipelines parallel to existing ones. This process is referred to as “looping” in the terminology of the industry. It is significantly cheaper than, for example, building completely new pipelines, since existing rights of way can be used simplifying the regulatory and the planning processes and, moreover, the additional encroachment on the environment is rather moderate.

In the literature, two different strands have been considered for modeling the looping problem and related expansion planning problems: the split-pipe and the discrete approach. In the first approach, the entire length of the pipeline is split into several segments each of which may have its own length and diameter (see, e.g., [9, 18, 41]), and the discrete approach, where the optimal diameters are chosen from a set of commercially available diameters (see, e.g., [4, 7, 17]).

In this paper, we consider expansion planning based on the split-pipe approach. More precisely, we use the formulation of Model E for the looping problem from [21], since the authors show that it performed best among all studied model formulations. In the following, we briefly explain the split-pipe approach and state its model formulation for the loop expansion planning problem.

Split-pipe approach The starting point is the concept of the so-called equivalent diameter. It allows to replace multiple pipelines in parallel by a single pipeline without changing any physical properties of the network, see, for example, [4, 7, 19] for two parallel pipelines in gas and water networks. For a generalization of the equivalent diameter to more than two pipelines in parallel, we refer to [20]. A type of equivalent diameter also exists for the case of multiple serial pipelines with no intermediate source or sink. When modeling split-pipes, it suffices to consider only those equivalent diameters resulting from parallel pipelines (multiple loops) and serial pipelines (split-pipe setting) that are not dominated by others with lower costs. These nondominated points together with their nondominated convex combinations form the so-called efficient frontier. This means that all points on the efficient frontier represent the cost minimal way of equipping a pipeline with a certain equivalent diameter. For more detailed information, we refer to Section 2 in [21].

Model formulation for the split-pipe approach Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a directed and connected graph with node set $\mathcal{V}$ and pipe set $\mathcal{A}$. A balanced demand vector $b \in \mathbb{R}^{|\mathcal{V}|}$ specifies nodal inflow and outflow values of the network. Here, we deal with a stationary model, this means that the overall network demand satisfies $\sum_{v \in \mathcal{V}} b_v = 0$. The physical state of the network is described by flow variables $x_a \in [\underline{x}_a, \bar{x}_a]$ for all arcs $a = (v, w) \in \mathcal{A}$ and nonnegative potential variables $\pi_v \in [\underline{\pi}_v, \bar{\pi}_v]$ at each node $v \in \mathcal{V}$. The arc flows arise from a friction-induced potential difference between the arcs’ adjacent nodes. In stationary models, it is a common approach to represent this interdependency by virtue of the Weymouth equation, see [38], with $\alpha = 2$ for gas
network operations and by virtue of Darcy–Weisbach with $\alpha = 2$ or Hazen–Williams with $\alpha = 1.852$ for water network operations, see [37], in any case taking the shape of Eq. (1a). Note that $\text{sgn}(x_a)$ indicates the direction of the arc flow. Given that, in general, bi-directional arc flow is possible, we assume that $x < 0$ and $x > 0$. Here, for each pipe $a \in \mathcal{A}$, positive variables $y_a \in [y_a, \overline{y}_a]$ and $c_a \geq 0$ model the efficient frontier of the split-pipe approach described above, where $y_a$ represents the physical impact of the expansion on the potential loss along arc $a$ with corresponding costs $c_a$ that can be assumed as proportional to the pipe length $L_a$. Lenz [20] shows that the best expansion candidates in the split-pipe setting are determined by the efficient frontier [Eq. (1b)], which is given by $k_a$-many nondominated points in the $(y_a, c_a)$-system. In the case that pipe $a$ is not looped, then by construction it follows $y_a = y_a$ with associated costs $c_a = 0$. Otherwise, $y_a > y_a$ and $c_a > 0$. Finally, as in classical network flow problems, Eq. (1c) models the flow conservation at every network node. Then, the model reads,

$$\min_{y, c, x, \pi} \sum_{a \in \mathcal{A}} L_a c_a$$

subject to

$$\pi_v - \pi_w = y_a \text{sgn}(x_a) |x_a|^\alpha \quad \forall a = (v, w) \in \mathcal{A},$$  

(1a)

$$c_a \geq r_{a,i} y_a + s_{a,i} \quad \forall a \in \mathcal{A} \forall i \in [k_a - 1],$$  

(1b)

$$\sum_{a \in \delta^+(v)} x_a - \sum_{a \in \delta^-(v)} x_a = b_v \quad \forall v \in \mathcal{V},$$

$$\pi_v \leq \pi_v \leq \overline{\pi}_v \quad \forall v \in \mathcal{V},$$

$$\underline{x}_a \leq x_a \leq \overline{x}_a \quad \forall a \in \mathcal{A},$$

$$y_a \leq y_a \leq \overline{y}_a \quad \forall a \in \mathcal{A},$$

$$c_a \geq 0 \quad \forall a \in \mathcal{V}. \quad (1c)$$

In this paper, we compute the convex hull of the constraint (1a) by studying the convex and concave envelope of the nonlinear function $(x, y) \mapsto y \text{sgn}(x) |x|^\alpha$ and then investigating its impact on the performance of expansion planning problem instances.

Finally, let us mention that the nonlinear nature of the expansion planning problem stimulated a number of different model formulations and solution approaches. For example, [3] introduced the linear programming gradient (LPG) method, a two-stage heuristic that alternates between solving a linear program with fixed flows in order to obtain pipeline diameters and modifying the flow values on the basis of a sensitivity analysis. André et al. [4] tackle the discrete version of the problem heuristically in two stages, where a first step solves the continuous relaxation in order to identify pipes to be looped and a second step determines discrete-valued diameters for these selected pipes based on branch-and-bound. Bragalli et al. [7] utilize a continuous reformulation of the cost function to solve the model directly with an MINLP solver. Raghunathan [27] proposes a disjunctive program together with a convex relaxation and then solves it to global optimality. Humpola [15] formulates a model and solves it to global optimality by using convex reformulation techniques and special tailored cuts. Borraz-Sánchez et al. [6] explicitly model forward and backward flow directions.
by using binary variables and present a solution approach based on a second-order cone relaxation.

3 Preliminaries on Convex Envelopes

In general, the derivation of convex and concave envelopes is a challenging task depending on the nonlinear constraint function and on the domain of the variables involved, see, for example, [14]. It can be shown that for a lower semicontinuous function \( f \) over a compact and convex domain \( D \), \( \min_{x \in D} f(x) = \min_{x \in D} \text{vex}_D[f](x) \) and hence computing the convex envelope is in general \( \mathcal{NP} \)-hard. Unless \( P = \mathcal{NP} \), no efficient method exists to compute convex envelopes of general nonlinear functions.

In order to build tight convex relaxations, convex envelopes have been widely studied for particular functions or classes of functions. For example, [25] constructs a linear relaxation over a box domain for the bilinear term \( xy \), which is proven to be the convex envelope by [2]. In the last decade, several approaches arose to convexify the bilinear term \( xy \) over special domains, see, e.g., [13, 23, 24, 26, 30]. The convex envelope of the fractional term \( y/x \) has been studied, e.g., in [33, 40]. Moreover, [33] computes the convex envelope of the function \( f(x, y) = g(y) x^2 \) over a rectangular domain \( D \subseteq \mathbb{R}^2 \), where \( f: D \to \mathbb{R} \) is convex in \( x \) and concave in \( y \). Even though this function is close to our constraint function \( f(x, y) = y \text{sgn}(x) |x|^\alpha \), this approach is not applicable in our setting, given that our function involves the nonconvex term \( \text{sgn}(x) |x|^\alpha \) instead of \( x^2 \).

Note that \( f(x, y) = y \text{sgn}(x) |x|^\alpha \) is twice differentiable only for \( \alpha > 2 \) and thus it is not possible to compute the convex envelope using techniques based on the Hessian, as done, for example, for \((n-1)\)-convex functions in [16]. In principle, when \( f \in C^2 \) (i.e., \( \alpha > 2 \)), we could generate valid tangential hyperplanes of function \( f \) to separate violated LP solutions without explicitly calculating the convex envelope, as done in [5] for twice continuously differentiable functions with fixed convexity behavior over a box. However, our analysis is valid for any \( \alpha > 1 \).

In the remainder of this section, we introduce some mathematical notation and necessary theoretical results that are required to derive \( \text{vex}_D(f) \). Even though our function is continuous, many of the following results hold for lower semicontinuous functions, which we define as follows.

**Definition 3.1** Let \( f: D \to \mathbb{R} \) with \( D \subseteq \mathbb{R}^n \). The *epigraph* of \( f \) is

\[
\text{epi}_D f = \{(x, \mu) | x \in D, \mu \in \mathbb{R}, \mu \geq f(x)\}.
\]

**Definition 3.2** A function \( f: D \to \mathbb{R} \) is *lower semicontinuous* (l.s.c.) if its epigraph is closed.

We continue by formally defining the convex envelope of a function.

**Definition 3.3** The *convex envelope* of a function \( f: D \to \mathbb{R} \) over a convex set \( D \subseteq \mathbb{R}^n \) is given by the tightest convex underestimator \( \eta: D \to \mathbb{R} \) of \( f \), defined pointwise
as
\[
\text{vex}_D[f](x) = \sup \{\eta(x) \mid \eta(y) \leq f(y) \text{ for all } y \in D, \eta \text{ convex}\}.
\]

The \textit{concave envelope} of a function \(f : D \to \mathbb{R}\) over \(D \subseteq \mathbb{R}^n\) is given by the tightest concave overestimator \(\eta : D \to \mathbb{R}\) of \(f\), defined pointwise as
\[
\text{cave}_D[f](x) = \inf \{\eta(x) \mid \eta(y) \geq f(y) \text{ for all } y \in D, \eta \text{ concave}\}.
\]

A geometrical characterization of the convex envelope is as follows.

**Theorem 3.1** [28, Theorem 5.3] \(\text{Let } E \text{ be any convex set in } \mathbb{R}^{n+1}, \text{ and let}
\]
\[
g(x) = \inf \{\mu \mid (x, \mu) \in E\}.
\]

Then, \(g : \mathbb{R}^n \to \mathbb{R}\) is a convex function.

The convex envelope of a l.s.c. function \(f\) corresponds to the function \(g\), obtained by applying Theorem 3.1 to the convex set \(\text{conv}(\text{epi}_D f)\). Therefore, it follows
\[
\text{vex}_D[f](x) = \min \{\mu \mid (x, \mu) \in \text{conv}(\text{epi}_D f)\},
\]
see also [28]. In order to determine the convex envelope of a l.s.c. function \(f\) at point \(x \in D\), its representation in (2) leads to the following minimization problem
\[
\text{vex}_D[f](x) = \min \sum_{k=1}^{n+1} \lambda_k f(x_k)
\]
\[
\sum_{k=1}^{n+1} \lambda_k x_k = x
\]
\[
\sum_{k=1}^{n+1} \lambda_k = 1
\]
\[
\lambda_k \geq 0, x_k \in D \text{ for } 1 \leq k \leq n + 1.
\]

Note that the \(n + 1\) is justified by \textit{Caratheodory’s theorem} [28, Theorem 17.1]. Problem (3) is nonconvex and nonlinear as the multipliers \(\lambda_k\) and the variables \(x_k\) are unknown. However, it is possible to reduce the degree of nonlinearity and non-
convexity of Problem (3) by studying the convex set \(\text{conv}(\text{epi}_D f)\) in more detail. In particular, we reduce the domain \(D\) by determining a subset \(A \subseteq D\), such that \(\text{conv}(\text{epi}_D f) = \text{conv}(\text{epi}_A f)\) holds.

**Definition 3.4** Let \(D\) be compact and convex subset of \(\mathbb{R}^n\), and let \(f : D \to \mathbb{R}\) be a l.s.c. function.
The generating set of the convex envelope of $f$ is defined by

$$G_D(f) := \{ x \in D \mid (x, \text{vex}_D[f](x)) \text{ is an extreme point of } \text{conv}(\text{epi}_D f) \}.$$ 

Given that a closed convex set containing no lines is characterized by the set of its extreme points and extreme directions [28, Theorem 18.5], we have the following result, see also [34].

**Theorem 3.2** Let $D$ be compact and convex subset of $\mathbb{R}^n$, $f : D \to \mathbb{R}$ be a l.s.c. function, and $A \subseteq D$. Then, $\text{conv}(\text{epi}_D f) = \text{conv}(\text{epi}_A f)$ if and only if $G_D(f) \subseteq A$.

Even though there exists no general formula to compute $G_D(f)$ of a nonconvex function $f$, [34] provides a condition that determines when a point $x \in D$ is not in $G_D(f)$.

**Corollary 3.1** [34, Corollary 5] Let $f : D \to \mathbb{R}$ be l.s.c. and $D \subseteq \mathbb{R}^n$ compact and convex. If for $x \in D$ there exists a line segment $l \subseteq D$ such that $x \in \text{ri}(l)$ and $f$ restricted to $l$ is concave, then $x \not\in G_D(f)$.

**4 Computing the Convex Envelope of $f(x, y) = y \, \text{sgn}(x) \, |x|^{\alpha}$**

In this section, we show how to compute the convex and concave envelope of $f : D \subseteq \mathbb{R}^2 \to \mathbb{R}$, with $f : (x, y) \mapsto y \, \text{sgn}(x) \, |x|^\alpha$, where $\alpha > 1$, over the box $D = [x, \overline{x}] \times [y, \overline{y}] \subseteq \mathbb{R}^2$. In the remainder of this paper, we assume $y > 0$, $\underline{x} < 0$ and $\overline{x} > 0$, as it typically occurs in practical applications, see Sect. 2. For a visualization of the nonconvex function $f$ with $\alpha = 2$, we refer to Fig. 5.

**4.1 Preliminaries about the Envelopes of $f(x, y) = y \, \text{sgn}(x) \, |x|^{\alpha}$**

Problem (3) for $f(x, y) = y \, \text{sgn}(x) \, |x|^\alpha$ over the domain $D = [x, \overline{x}] \times [y, \overline{y}] \subseteq \mathbb{R}^2$ reads as

$$\text{vex}_D[f](x, y) = \min \sum_{k=1}^{3} 3 \lambda_k y_k \, \text{sgn}(x_k) \, |x_k|^{\alpha},$$

$$\sum_{k=1}^{3} \lambda_k \, x_k = x,$$

$$\sum_{k=1}^{3} \lambda_k \, y_k = y,$$

$$\sum_{k=1}^{3} \lambda_k = 1,$$

$$\lambda_k \geq 0, (x_k, y_k) \in D \quad \forall 1 \leq k \leq 3.$$
Fig. 1 Illustration of Proposition 4.1 for the one-dimensional functions $f_y(x) = y \sgn(x)x^2$ (blue, dotted) and $g_y(x) := -f_y(-x) = -y \sgn(-x)(-x)^2$ (red, dashed) with $f_y : D := [x, \bar{x}] \to \mathbb{R}$ and $g_y : D_g := [-\bar{x}, -x] \to \mathbb{R}$ for fixed $y$, where $\mathrm{vex}_{D_g}[g_y](x)$ (red, solid) is the convex envelope of $g_y$ and $-\mathrm{vex}_{D_g}[g_y](-x) = \mathrm{cave}_D[f](x)$ (blue, solid) the concave envelope of $f_y$.

Note that the above problem has nine variables ($x_k, y_k, \lambda_k$ for $1 \leq k \leq 3$). In the next section, we show how to reduce this problem to a one-dimensional problem.

Here, the structure of the function $f(x, y) = y \sgn(x)|x|^\alpha$ allows us to deduce both envelopes from each other, see Fig. 1. The function $f(x, y) = y \sgn(x)|x|^\alpha$ is odd in $x$ when $y$ is fixed. This allows us to retrieve the concave envelope from the convex envelope, and thus, from now on, we restrict ourselves to the derivation of the convex envelope.

**Proposition 4.1** Let $f : \mathbb{R} \times [y, \bar{y}] \to \mathbb{R}$ be an odd function in $x$, i.e., $f(x, y) = -f(-x, y)$ and let $D := [x, \bar{x}] \times [y, \bar{y}]$. Let $D_g := [-\bar{x}, -x] \times [y, \bar{y}]$ and $g : D_g \to \mathbb{R}$ be given by $g(x, y) := -f(-x, y)$. Then, the concave envelope of $f$ over $D$ is given by $\mathrm{cave}_D[f](x, y) = -\mathrm{vex}_{D_g}[g](-x, y)$.

**Proof** By definition of $g$ and $\mathrm{vex}_{D_g}[g](x, y)$, we have that

$$\mathrm{vex}_D[-f](-x, y) = \mathrm{vex}_{D_g}[g](x, y) \quad \text{with} \quad (x, y) \in D_g.$$ 

Thus,

$$\mathrm{cave}_D[f](x, y) = -\mathrm{vex}_{D_g}[g](-x, y) \quad \text{with} \quad (x, y) \in D.$$ 

\[ \square \]
4.2 Problem Reduction by Using the Generating Set

To simplify Problem (3), we use Corollary 3.1 to find a set $A \subseteq D$ such that $G_D(f) \subseteq A$ for $f(x, y) = y \text{sgn}(x) |x|^\alpha$.

First, notice that after fixing $x$ to a value in $[\underline{x}, \overline{x}]$, the function $f(x, y)$ is linear and thus concave over the line segment $l = \{(x, y) | y \in [\underline{y}, \overline{y}]\}$. Then, Corollary 3.1 implies that all points in $\text{ri}(l) = \{(x, y) | y \in (\underline{y}, \overline{y})\}$ are not in $G_D(f)$. Thus,

$$G_D(f) \subseteq \{(x, y), \underline{x} \leq x \leq \overline{x}\} \cup \{(x, \overline{y}), \underline{x} \leq x \leq \overline{x}\}.$$

Let $D_y := \{(x, y) | \underline{x} \leq x \leq \overline{x}\}$ and $D_\overline{y} := \{(x, \overline{y}) | \underline{x} \leq x \leq \overline{x}\}$. Given that $G_D(f) \subset D_y \cup D_\overline{y}$, Theorem 3.2 allows us to express the convex hull of $\text{epi}_D f$ as

$$\text{conv}(\text{epi}_D f) = \text{conv}(\text{epi}_{D_y \cup D_\overline{y}} f) = \text{conv}(\text{epi}_{D_y} f \cup \text{epi}_{D_\overline{y}} f) = \text{conv}(\text{conv}(\text{epi}_{D_y} f) \cup \text{conv}(\text{epi}_{D_\overline{y}} f)). \quad (4)$$

The convex hulls $\text{conv}(\text{epi}_{D_y} f)$ and $\text{conv}(\text{epi}_{D_\overline{y}} f)$ correspond to the epigraphs of the convex envelopes of the one-dimensional functions $f_y(x) := f(x, y)$ and $f_\overline{y}(x) := f(x, \overline{y})$. The following proposition introduces these one-dimensional convex envelopes, see Fig. 2 for an illustration.

**Proposition 4.2** [35, Section 7.5.2] The convex envelope of the function $f_y : [\underline{x}, \overline{x}] \to \mathbb{R}$ is given by $f_y(x) = y \text{sgn}(x) |x|^\alpha$, where $\underline{x} < 0 < \overline{x}$, $\alpha > 1$, and $y > 0$ is given by

$$\varphi_y(x) = \begin{cases} yx^\alpha & x \geq \beta \underline{x} \\ \alpha y^{\alpha-1}x + (1 - \alpha)\gamma^\alpha y & x < \beta \underline{x}. \end{cases} \quad (5)$$

where $\gamma = \min\{\overline{x}, \beta \underline{x}\}$ and $\beta$ is the unique negative root of $(\alpha - 1)(-\beta)^\alpha + \alpha(-\beta)^{\alpha-1} - 1$.

**Remark 4.1** The function $(x, y) \mapsto \varphi_y(x)$ is not the convex envelope of $f(x, y)$ as can be seen from Fig. 6 (middle). This function is not even convex! \(\square\)

Given that $\text{conv}(\text{epi}_{D_y} f) = \text{epi}_{D_y} \varphi_y$ and $\text{conv}(\text{epi}_{D_\overline{y}} f) = \text{epi}_{D_\overline{y}} \varphi_\overline{y}$, we can further rewrite (4) to

$$\text{conv}(\text{epi}_D f) = \text{conv}\left(\text{epi}_{D_y} \varphi_y \cup \text{epi}_{D_\overline{y}} \varphi_\overline{y}\right).$$

Therefore, we have

$$\text{vex}_D[f](x, y) = \min\{\mu | (x, y, \mu) \in \text{conv}\left(\text{epi}_{D_y} \varphi_y \cup \text{epi}_{D_\overline{y}} \varphi_\overline{y}\right)\}. \quad (\lozenge)$$
The convex envelope \( \varphi_y \) (red) of the one-dimensional function \( f_y \) (blue) depends on the relation of the values \( \beta_x \) and \( x \). In the case of \( \beta_x < x \), the convex envelope is given by the secant between the points \( (x, f_y(x)) \) and \( (\beta_x, f_y(\beta_x)) \) for \( x \in [\beta_x, x] \) and the function \( f_y(x) \) itself for \( x \in [\beta_x, \beta_x] \). Otherwise, if \( \beta_x > x \), then \( \text{vex}_{[\beta_x, x]}[f_y](x) \) is given by the secant between \( (x, f_y(x)) \) and \( (\beta_x, f_y(\beta_x)) \).

Equivalently,

\[
\text{vex}_D[f](x, y) = \min_{x_1, x_2, \lambda} (1 - \lambda)\varphi_y(x_1) + \lambda\varphi_y(x_2)
\]

s.t. \( (1 - \lambda)x_1 + \lambda x_2 = x \)

\( (1 - \lambda)y + \lambda \bar{y} = y \)

\( \lambda \in [0, 1], x_1, x_2 \in [x, \beta_x] \).

Notice that \( \text{vex}_D[f](x, y) = \varphi_y(x) \) and \( \text{vex}_D[f](x, \bar{y}) = \varphi_y(x) \). Hence, we just need to determine the convex envelope of \( f \) for \( y \in (\underline{y}, \bar{y}) \). In the following, we assume that \( y \in (\underline{y}, \bar{y}) \).

### 4.3 Reduction to a One-Dimensional Optimization Problem

We can further reduce the optimization Problem (6) to a one-dimensional problem. Firstly, (6b) enables us to fix the multiplier \( \lambda \)

\[
\lambda_y := \lambda = \frac{y - \underline{y}}{\bar{y} - \underline{y}} \in (0, 1).
\]
An interpretation of function $T_{x,y}$ on $D = [x, \bar{x}] \times (y, \bar{y})$. The set $\{(x, y): T_{x,y}(a) \leq b\}$ with $x \leq a \leq b \leq \bar{x}$ represents the gray-shaded region in $D$, and $\{(x, y): b \leq T_{x,y}(a)\}$ corresponds to the region in white in $D$. Both regions are separated by the line segment that connects the points $(a, y)$ and $(b, y)$. As we will see, the value of the convex envelope will depend on conditions of the form $T_{x,y}(a) \leq b$ for appropriate values of $a$ and $b$. Thus, the reader can use this figure to get an idea on how the position of the point $(x, y) \in D$ is related to the expression of the convex envelope.

Given that $\lambda$ is fixed, we can use (6a) to define $x_2$ in terms of $x_1$. For this, we define the functions $t_{x,y}, T_{x,y}: [x, \bar{x}] \to \mathbb{R}$ given by

$$t_{x,y}(z) = \frac{x - (1 - \lambda_y)z}{\lambda_y}, \quad \text{and}$$

$$T_{x,y}(z) = \frac{x - \lambda_y z}{1 - \lambda_y}. \quad (7)$$

These functions satisfy $x_2 = t_{x,y}(x_1)$ and $x_1 = T_{x,y}(x_2)$ for every feasible solution $x_1, x_2$ of Problem (6). Both functions are affine linear, strictly decreasing and inverse to each other. For an interpretation of $T_{x,y}$, see Fig. 3. Note that $t_{x,y}$ and $T_{x,y}$ are well defined, since we assume $\lambda_y \in (0, 1)$, as described above.

Projecting out $x_2$ in Problem (6) yields a one-dimensional problem, which we can express with the help of $t_{x,y}$ and $T_{x,y}$ as

$$\min (1 - \lambda_y)\varphi_y(x_1) + \lambda_y \varphi(t_{x,y}(x_1))$$

s.t. $\max \{x, T_{x,y}(\bar{x})\} \leq x_1 \leq \min \{\bar{x}, T_{x,y}(x)\}. \quad (9)$

The expression for the bounds come from combining both bounds of $x_1$, namely $x \leq x_1 \leq \bar{x}$ and $T_{x,y}(\bar{x}) \leq x_1 \leq T_{x,y}(x)$. The latter bounds come from $x \leq x_2 \leq \bar{x}$, which after projecting out $x_2$ becomes $x \leq t_{x,y}(x_1) \leq \bar{x}$. Applying the function $T_{x,y}$ to the inequality yields $T_{x,y}(\bar{x}) \leq x_1 \leq T_{x,y}(x)$.

Let $\underline{x}_1 := \max \{x, T_{x,y}(\bar{x})\}, \bar{x}_1 := \min \{\bar{x}, T_{x,y}(x)\}$, and let $F_{x,y}: [\underline{x}_1, \bar{x}_1] \to \mathbb{R}$ be

$$F_{x,y}(x_1) = (1 - \lambda_y)\varphi_y(x_1) + \lambda_y \varphi(t_{x,y}(x_1)). \quad (10)$$
Then, Problem (9) is equivalent to

\[
\min_{x_1, y} F_{x_1, y}(x_1)
\]
\[
\text{s.t. } x_1 \leq x_1 \leq \overline{x}_1.
\]

(11)

Remark 4.2 Problem (11) allows for a geometrical interpretation as illustrated in Fig. 4:
For a given point \((x, y)\) \(\in D\), Problem (11) corresponds to finding the points \((x_1, y)\) and \((t_{x,y}(x_1), y)\) such that the line segment between these points contains \((x, y)\), and the line segment with endpoints \((x_1, y, \varphi_y(x_1))\) and \((t_{x,y}(x_1), y, \varphi_y(t_{x,y}(x_1)))\) has the lowest height at \((x, y)\). Note that the compact formulation of Problem (11) only contains \(x_1\) as a variable, where constraint (7) allows us to recover \(x_2 = t_{x,y}(x_1)\).

At this point, we can already compute the convex envelope for a simple case.

Proposition 4.3 If \(\overline{x} < \beta \underline{x}\), then \(\text{vex}_D[f](x, y) = F_{x_1, y}(\min \{\overline{x}, T_{x_1, y}(\underline{x})\})\).

Proof When \(\overline{x} < \beta \underline{x}\) holds, then \(\varphi_y(x) = \alpha \overline{x}^{\alpha - 1} x y + (1 - \alpha) \underline{x}^\alpha y\). Therefore, \(F_{x_1, y}\) is a linear function and (11) reduces to minimizing a linear function over an interval. A simple computation shows that the coefficient of \(x_1\) in \(F_{x_1, y}(x_1)\) is negative, and so the minimum is achieved at the upper bound, i.e., \(\text{vex}_D[f](x, y) = F_{x_1, y}(\min \{\overline{x}, T_{x_1, y}(\underline{x})\})\).
4.4 Properties of the Objective Function

The objective function of Problem (11), $F_{x,y}$, is convex as it is a convex combination of convex functions. Note that the function $\varphi_y(t_{x,y}(\cdot))$ is convex since it is the composition of a convex ($\varphi_y$) with an affine linear function ($t_{x,y}$).

The following proposition characterizes the optimal solution for a large class of problems of a form similar to Problem (11).

**Proposition 4.4** Suppose that a function $F: \mathbb{R} \rightarrow \mathbb{R}$ is convex and has a global minimum $z^* \in \mathbb{R}$. Then, the solution of $\min \{ F(z) : a \leq z \leq b \}$ is given by

$$F(\text{mid}(a, b, z^*)),$$

where $\text{mid}(a, b, z^*)$ selects the middle value between $a$, $b$, $z^*$.

**Proof** The claim follows directly by comparing $F(a)$, $F(b)$ and $F(z^*)$. If $z^*$ is within the bounds, i.e., $a \leq z^* \leq b$, then $F(\text{mid}(a, b, z^*)) = F(z^*)$. If $z^* \leq a$, then the convexity of $F$ implies that $F$ is increasing in $[z^*, \infty)$ and so $F(a) = F(\text{mid}(a, b, z^*))$ is the minimum. The argument is analogous for the case $z^* \geq b$. $\Box$

In order to apply Proposition 4.4, we need to extend $F_{x,y}$ from $x_1 \in [\underline{x}_1, \overline{x}_1]$ to $x_1 \in \mathbb{R}$ and show that it has a global minimum. The extension is immediate, given that all the functions involved can be evaluated in $\mathbb{R}$. To show that $F_{x,y}$ has a global minimum over $\mathbb{R}$, we show that the sublevel sets of $F_{x,y}$ are bounded. This follows from the following proposition.

**Proposition 4.5** The function $F_{x,y}: \mathbb{R} \rightarrow \mathbb{R}$, given by Eq. (10) with $\alpha > 1$, is coercive, i.e., it satisfies

$$\lim_{x_1 \to +\infty} F_{x,y}(x_1) = \infty \quad \text{and} \quad \lim_{x_1 \to -\infty} F_{x,y}(x_1) = \infty.$$

**Proof** We compute the limit when $x_1$ goes to $+\infty$. The case when $x_1$ goes to $-\infty$ is similar. By definition,

$$F_{x,y}(x_1) = (1 - \lambda_y)\varphi_{\underline{y}}(x_1) + \lambda_y\varphi_{\gamma}(t_{x,y}(x_1)).$$

For every large enough $x_1$, we have $x_1 \geq \beta \underline{x}$ and $t_{x,y}(x_1) < \beta \underline{x}$. Thus,

$$F_{x,y}(x_1) = (1 - \lambda_y)\underline{x}_{1}^{\alpha} + \lambda_y\left(\alpha(\beta \underline{x})^{\alpha-1}t_{x,y}(x_1)\underline{y} + (1 - \alpha)(\beta \underline{x})^{\alpha} \underline{y}\right).$$

This can be rewritten as $ax_1^{\alpha} + bx_1 + c$, where $a > 0$, $b < 0$ (since $t_{x,y}$ is strictly decreasing and linear), and $c \in \mathbb{R}$. Given that $\alpha > 1$, the sign of $a$ determines the behavior of $F_{x,y}$ at infinity; hence, $\lim_{x_1 \to +\infty} F_{x,y}(x_1) = \infty$. $\Box$

Figure 5 visualizes function $F_{x,y}(x_1)$ for different pairs of points $(x, y)$. Springer
Fig. 5 An illustration of function $F_{x,y}(x_1)$ for different pairs of values $(x, y) \in \{-200, 200\} \times \{0.1, 0.5\}$ and \(y = 1/100, y = 1\). The function $F_{x,y}$ is continuous, convex and coercive

4.5 Solution to Problem (11)

With these properties of $F_{x,y}$, we are ready to compute its global minimum over $\mathbb{R}$. To this end, we use that $F_{x,y}$ is differentiable, since $\varphi_y \in C^1$ for $\alpha > 1$, and solve $F'_{x,y}(x_1) = 0$. We have

$$F'_{x,y}(x_1) = (1 - \lambda_y)\varphi'_y(x_1) + \lambda_y \varphi'_y(t_{x,y}(x_1)) t'_{x,y}(x_1) \tag{7}$$

Then, given the piecewise nature of $\varphi_y$, see (5), we get

$$F'_{x,y}(x_1) = (1 - \lambda_y)\begin{cases} \alpha(\beta_x)^{\alpha-1}y - \alpha(\beta_x)^{\alpha-1}\bar{y} & \text{if } x_1 \in LL \\ \alpha(\beta_x)^{\alpha-1}y - \alpha t_{x,y}(x_1)^{\alpha-1}\bar{y} & \text{if } x_1 \in LR \\ \alpha x_1^{\alpha-1}y - \alpha(\beta_x)^{\alpha-1}\bar{y} & \text{if } x_1 \in RL \\ \alpha x_1^{\alpha-1}y - \alpha t_{x,y}(x_1)^{\alpha-1}\bar{y} & \text{if } x_1 \in RR, \end{cases} \tag{12}$$

where

$$LL = \{x_1 \in \mathbb{R}: x_1 \leq \beta_x, t_{x,y}(x_1) \leq \beta_x\},$$

$$LR = \{x_1 \in \mathbb{R}: x_1 \leq \beta_x, t_{x,y}(x_1) \geq \beta_x\},$$

$$RL = \{x_1 \in \mathbb{R}: x_1 \geq \beta_x, t_{x,y}(x_1) \leq \beta_x\},$$

$$RR = \{x_1 \in \mathbb{R}: x_1 \geq \beta_x, t_{x,y}(x_1) \geq \beta_x\}.$$

To solve $F'_{x,y}(x_1) = 0$, we restrict $x_1$ to be in $LL, LR, RL$ or $RR$.  

\(\copyright\) Springer
Case $x_1 \in LL$: From (12), it follows that $F'_{x,y}(x_1) = 0$ if and only if
\[
\alpha(\beta x)^{\alpha-1} \gamma = \alpha(\beta x)^{\alpha-1} \bar{y}.
\]
Given that $y < \bar{y}$, it follows $F'_{x,y}(x_1) \neq 0$ for all $x_1 \in LL$. Thus, the global minimum is not in $LL$.

Case $x_1 \in LR$: The solution of $F'_{x,y}(x_1) = 0$ is given by
\[
x_{lr} := T_{x,y} \left( \beta x \left( y / \bar{y} \right)^{\frac{1}{\alpha-1}} \right).
\]

Note, however, that $x_{lr} \notin LR$, as $t_{x,y}(x_{lr}) < \beta x$. Thus, the global minimum is not in $LR$.

Case $x_1 \in RL$: The solution of $F'_{x,y}(x_1) = 0$ is given by
\[
x_{rl} := \beta x \left( \bar{y} / y \right)^{\frac{1}{\alpha-1}}.
\]

The point $x_{rl}$ is in $RL$ if and only if $x_{rl} \geq \beta x$ and $t_{x,y}(x_{rl}) \leq \beta x$. Given that $\beta, x < 0$, we have that $x_{rl} > \beta x$. Thus, the global minimum is $x_{rl}$ if and only if $t_{x,y}(x_{rl}) \leq \beta x$. Otherwise, the global minimum must be in the next case.

Case $x_1 \in RR$: The solution of $F'_{x,y}(x_1) = 0$ is given by
\[
x_{rr} := x \frac{\bar{y}^{\frac{1}{\alpha-1}}}{\lambda_y \bar{y}^{\frac{1}{\alpha-1}}} + (1 - \lambda_y) \bar{y}^{\frac{1}{\alpha-1}}.
\]

From the above case distinction, it follows that the optimal solution is $x_{rl}$ if $t_{x,y}(x_{rl}) \leq \beta x$, otherwise it is $x_{rr}$. Therefore, from Proposition 4.4, we conclude that
\[
\text{vex}_D[f](x, y) = \begin{cases} F_{x,y}(\text{mid}(\underline{x}_1, \bar{x}_1, x_{rl})) & \text{if } t_{x,y}(x_{rl}) \leq \beta x \\ F_{x,y}(\text{mid}(\bar{x}_1, \underline{x}_1, x_{rr})) & \text{else.} \end{cases}
\]

Figure 6 shows a visualization of $y \text{sgn}(x) |x|^\alpha$ and its convex envelope.
4.6 Simplifying the Convex Envelope

We now derive a representation of $\text{vex}_D[f](x, y)$ more compact than (15).

**Theorem 4.1** The convex envelope of $f(x, y) = y \text{sgn}(x)|x|^\alpha$ over $D = [\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$ is given by

$$\text{vex}_D[f](x, y) = F_{x,y}(\min \{\bar{x}, T_{x,y}(x), \max\{x_{rl}, x_{rr}\}\}).$$  \hspace{1cm} (16)

**Proof** To prove the claim, we rewrite (15). We first show that

$$t_{x,y}(x_{rl}) \leq \beta \underline{x} \iff x_{rl} \geq x_{rr},$$  \hspace{1cm} (17)

where $x_{rl}$ and $x_{rr}$ are given in (13) and (14).

Then, we can rewrite (15) as

$$\text{vex}_D[f](x, y) = F_{x,y}(\text{mid}(\underline{x}_1, \bar{x}_1, \max\{x_{rl}, x_{rr}\})).$$

Then, we show that

$$\bar{x}_1 \leq \max\{x_{rl}, x_{rr}\}.$$  \hspace{1cm} (18)

This implies that $\text{mid}(\underline{x}_1, \bar{x}_1, \max\{x_{rl}, x_{rr}\}) = \min\{\bar{x}_1, \max\{x_{rl}, x_{rr}\}\} = \min\{\bar{x}, T_{x,y}(x), \max\{x_{rl}, x_{rr}\}\}$, which proves the result.

To prove (17) and (18), define $G_{x,y} : \mathbb{R} \to \mathbb{R}$ as

$$G_{x,y}(z) := z - t_{x,y}(z)\left(\frac{\bar{y}}{\underline{y}}\right)^{\frac{1}{\alpha - 1}},$$

and notice that $G_{x,y}(x_{rr}) = 0$ and $G_{x,y}$ is strictly increasing.

The equivalence (17) follows from the following computation,

$$x_{rl} \leq x_{rr} \iff G_{x,y}(x_{rl}) \leq G_{x,y}(x_{rr}) \iff G_{x,y}(x_{rl}) \leq 0 \iff x_{rl} - t_{x,y}(x_{rl})\left(\frac{\bar{y}}{\underline{y}}\right)^{\frac{1}{\alpha - 1}} \leq 0 \iff x_{rl} \leq t_{x,y}(x_{rl})\left(\frac{\bar{y}}{\underline{y}}\right)^{\frac{1}{\alpha - 1}} \iff \beta \underline{x} \leq t_{x,y}(x_{rl}).$$

To prove (18), it is enough to show $\bar{x}_1 \leq x_{rr}$. Recall that $\bar{x}_1 = \max\{\bar{x}, T_{x,y}(\bar{x})\}$.

The inequality $\bar{x} \leq x_{rr}$ follows from the definition of $x_{rr}$ and $\lambda_y, y, \bar{y} > 0$.

Finally, let us show $T_{x,y}(\bar{x}) \leq x_{rr}$. This follows from the following computation,

$$T_{x,y}(\bar{x}) \leq x_{rr} \iff G_{x,y}(T_{x,y}(\bar{x})) \leq 0.$$
\[ \iff \quad T_{x,y}(\bar{x}) \leq \bar{x} \left( \frac{y}{y} \right)^{\frac{1}{\alpha-1}} \]

\[ \iff \quad x \leq (1 - \lambda_y)\bar{x} \left( \frac{y}{y} \right)^{\frac{1}{\alpha-1}} + \lambda_y \bar{x}, \]

and the fact that \( x \leq \bar{x} \leq (1 - \lambda_y)\bar{x} \left( \frac{y}{y} \right)^{\frac{1}{\alpha-1}} + \lambda_y \bar{x}. \)

\[ \Box \]

**Remark 4.3** From (16), we see that if \( x_{rl} \geq \bar{x} \), then

\[ \text{vex}_D[f](x, y) = F_{x,y}(\min \{ \bar{x}, T_{x,y}(x) \}) = F_{x,y}(\bar{x}_1). \]

The condition \( x_{rl} \geq \bar{x} \) depends only on the bounds, and not on the given point \((x, y)\). Therefore, if the bounds are such that \( x_{rl} \geq \bar{x} \), then we can further reduce the description of the convex envelope as above. \[ \Box \]

In summary, we derived an analytic solution for the convex envelope of the function

\[ f(x, y) = y \, \text{sgn}(x) \, |x|^\alpha, \]

with \( \alpha > 1 \), which is stated in (16). To evaluate the convex envelope at a point \((x, y)\) only requires the computation of the function \( F_{x,y} \) at the minimum between \( \bar{x}_1, x_{rl} \) and \( x_{rr} \).

## 5 Convexification of \( y \, \text{sgn}(x) \, |x|^\alpha \) in SCIP

In this section, we compare the convex underestimator induced by SCIP to the convex envelope of \( y \, \text{sgn}(x) \, |x|^\alpha \). To this end, consider the epigraph of \( y \, \text{sgn}(x) \, |x|^\alpha \),

\[ \{(x, y, \theta) \in D \times \mathbb{R} : \theta \geq y \, \text{sgn}(x) \, |x|^\alpha\}. \quad (19) \]

By virtue of Theorem 3.1, any convex relaxation of (19) induces a convex underestimator of \( y \, \text{sgn}(x) \, |x|^\alpha \). Therefore, we construct the convex underestimator of \( y \, \text{sgn}(x) \, |x|^\alpha \) induced by SCIP’s convex relaxation of (19) and compare it to the convex envelope of \( y \, \text{sgn}(x) \, |x|^\alpha \).

SCIP, as other MINLP solvers, uses a standard reformulation procedure that allows to generate a convex relaxation for factorable MINLPs. The idea is to decompose a factorable constraint function into simpler functions for which convex underestimators or even convex envelopes are known. For general information about this reformulation method, we refer to [32], and for a description of how it works in SCIP to [35].

The decomposition is achieved by introducing additional variables forming, thus, an extended formulation of the original problem. In our case, SCIP decomposes the constraint \( \theta \geq y \, \text{sgn}(x) \, |x|^\alpha \) by introducing an auxiliary variable \( z \) into

\[ z = \text{sgn}(x) \, |x|^\alpha, \]

\[ \theta \geq yz. \]

Then, SCIP builds a relaxation by computing the convex envelopes of \( f_1(x) = \text{sgn}(x) \, |x|^\alpha \) and \( f_2(y, z) = yz \). Note that by Proposition 4.2, the convex envelope
of $f_1$ is $\varphi_y$ with $y = 1$, which we denote by $\varphi_1$. Moreover, the convex envelope of $f_2$ is given by the well-known McCormick inequalities [25],

$$
\psi(y, z) := \max \{yz + yz - yz, yz + yz - yz\},
$$

where $z = \min_{x \in [\underline{x}, \overline{x}]} f_1(x)$ and $\overline{z} = \max_{x \in [\underline{x}, \overline{x}]} f_1(x)$.

The convex relaxation of (19) constructed by SCIP is

$$
C = \text{Proj}_{x, y, \theta} \{(x, y, z, \theta) : z \geq \varphi_1(x), \theta \geq \psi(y, z)\}.
$$

Hence, the underestimator induced by $C$ is

$$
\phi_S(x, y) := \min_{\theta, z} \{\theta : z \geq \varphi_1(x), \theta \geq \psi(y, z)\}. \tag{20}
$$

Given that $\psi$ is increasing with respect to $z$ for every $y$, an optimal solution of Problem (20) satisfies $z = \varphi_1(x)$. Therefore, the convex underestimator induced by SCIP is $\phi_S(x, y) = \psi(y, \varphi_1(x))$.

Figure 7 shows the difference between $\phi_S$ and $\text{vex}_D[f]$ over $D$.

From the figure, we can see that, for $D = [-100, 100] \times [0.01, 1]$, $\phi_S$ and $\text{vex}_D[f]$ coincide in $\{(x, y) \in D : T_{x,y}(x) \leq \beta_S\}$, see also Fig. 3. Using $D = [-100, 20] \times [0.01, 1]$ instead would show that $\phi_S$ and $\text{vex}_D[f]$ coincide in all of $D$. The next proposition explains this behavior.

**Proposition 5.1** If $\beta_S \geq \overline{x}$, then $\text{vex}_D[f](x, y) = \phi_S(x, y)$ for all $(x, y) \in D$.

**Proof** Proposition 4.3 implies that $\text{vex}_D[f](x, y) = F_{x,y}(\min \{\overline{x}, T_{x,y}(x)\})$. As $\beta_S \geq \overline{x}$, we have that $\min \{\overline{x}, T_{x,y}(x)\} \leq \beta_S$ and $t_{x,y}(\min \{\overline{x}, T_{x,y}(x)\}) \leq \beta_S$. Therefore,
in \( F_{x,y}(\min \{ \bar{x}, T_{x,y}(x) \}) \) we have that \( \varphi_y \) and \( \varphi_{\overline{x}} \) are evaluated on points smaller than \( \beta \bar{x} \), and so they are affine linear.

Assume that \( \min \{ \bar{x}, T_{x,y}(x) \} = T_{x,y}(x) \). Notice that \( T_{x,y}(x) \) is an affine combination of \( x \) and \( \bar{x} \) as \( T_{x,y}(x) = \frac{1}{1-\lambda_y} x + \frac{-\lambda_y}{1-\lambda_y} \bar{x} \) and \( \frac{1}{1-\lambda_y} + \frac{-\lambda_y}{1-\lambda_y} = 1 \). As \( \varphi_{\overline{x}} \) is affine linear, we have that \( (1-\lambda_y) \varphi_{\overline{x}}(T_{x,y}(x)) = \varphi_{\overline{x}}(x) - \lambda_y \varphi_y(x) \).

Therefore, \( F_{x,y}(T_{x,y}(x)) = \varphi_y(x) - \lambda_y \varphi_y(x) + \lambda_y \varphi_{\overline{x}}(x) \).

From \( \varphi_y(x) = y \varphi_1(x) \) and the definition of \( \lambda_y \), we obtain

\[
F_{x,y}(T_{x,y}(x)) = y \varphi_1(x) + y \varphi_1(x) - y \varphi_1(x).
\]

Notice that \( z = \varphi_1(x) \) (as \( \varphi_1 \) is the convex envelope of \( f_1 \)) and so

\[
F_{x,y}(T_{x,y}(x)) = \max \{ yz, y \varphi_1(x) - yz \} = \phi_5(x, y).
\]

However, \( F_{x,y}(T_{x,y}(x)) \) is the convex envelope of \( y \operatorname{sgn}(x) |x|^\alpha \), while \( \phi_5(x, y) \) is just a convex underestimator, which implies that \( \phi_5(x, y) \leq F_{x,y}(T_{x,y}(x)) \). From here, we conclude that \( \phi_5(x, y) = F_{x,y}(T_{x,y}(x)) \) as we wanted to show.

The case of \( \min \{ \bar{x}, T_{x,y}(x) \} = \bar{x} \) follows from a similar argument. \( \square \)

We now explain the behavior from Fig. 7.

**Proposition 5.2** Assume that \( \beta \bar{x} < \bar{x} \) and let \( A := \{(x, y) \in D : T_{x,y}(x) \leq \beta \bar{x} \} \). Then, for every \((x, y) \in A\),

\[
\phi_5(x, y) = \text{vex}_D[f](x, y).
\]

Furthermore, \( \text{vex}_D[f] \) is linear in \( A \).

**Proof** Given that \( \beta \bar{x} < \bar{x} \) and \( T_{x,y}(x) \leq \beta \bar{x} \), we have that \( \min \{ \bar{x}, T_{x,y}(x) \} = T_{x,y}(x) \) and \( t_{x,y}(\min \{ \bar{x}, T_{x,y}(x) \}) = \bar{x} \leq \beta \bar{x} \). Thus, we can apply the same reasoning as the one given in the above proof of Proposition 5.1. \( \square \)

**Remark 5.1** A similar behavior can be observed between the concave overestimator induced by SCIP and the concave envelope of \( y \operatorname{sgn}(x) |x|^\alpha \).

\( \square \)

### 6 Computational Study

In this section, we present a computational study that investigates the impact of the presented convex envelope on the performance of the expansion planning problem (see Sect. 2). For this, we extend SCIP with the convex envelope and compare it against default SCIP. We conduct two experiments to answer the following questions:

1. **ROOTGAP**: How much gap can be additionally closed in the root node of the branch-and-bound tree when using the envelopes of \( y \operatorname{sgn}(x) |x|^\alpha \) with aggressive separation settings?
2. **TREE**: To which extent do the presented envelopes affect the performance of SCIP, both in running time and solvability?
6.1 Experimental Setup

In order to measure the impact of the convex envelope over the standard relaxation in terms of improving the dual bound, we use an aggressive emphasis setting for the separation in the ROOTGAP experiment. To reduce the impact of side effects, we disable restarts, \(^1\) all primal heuristics, \(^2\) and we give the value of the best known primal solution.

In contrast to the ROOTGAP experiment, the TREE experiment compares SCIP default against SCIP with the additional separation routine that generates gradient cuts to the envelopes of \(f\) in every local node during the entire tree search. We remark that these gradient cuts are, in general, only locally valid since we compute them using the local bounds of the node.

In both experiments, the gradient cuts for the constraint \(y \operatorname{sgn}(x) |x|^2 = \pi_v - \pi_w\) using the convex and concave envelope read, respectively,

\[
\operatorname{vex}_D[f](x_0, y_0) + \nabla \operatorname{vex}_D[f](x_0, y_0)^T \left(\frac{x - x_0}{y - y_0}\right) \leq \pi_v - \pi_w
\]

and

\[
\operatorname{cave}_D[f](x_0, y_0) + \nabla \operatorname{cave}_D[f](x_0, y_0)^T \left(\frac{x - x_0}{y - y_0}\right) \geq \pi_v - \pi_w.
\]

6.2 Test Sets

The computational study is carried out on three test sets taken from \([20]\): Belgium, GasLib-40 and Circuit rank. These test sets are based upon different gas networks, demand scenarios and expansion candidates. In total, all test sets contain 6500 instances, with Belgium and GasLib-40 having each 2000 instances, and Circuit rank having 2500 instances. As we use gas instances, the exponent \(\alpha\) is set to \(\alpha = 2\) in Eq. (1a).

Networks The underlying networks of the test sets vary in size and structure. Here, we consider the rather small and almost tree-shaped Belgian gas network from the GAMS model library\(^3\) and the larger GasLib-40 network from \([29]\). In general, optimization problems get more challenging to solve on potential-driven networks that have a higher circuit rank, which is the number \(|A| - |V| + 1\), because the existence of cycles leads to more complex patterns of flow directions, see \([31]\) and the references therein. To also account for this type of computational difficulty, the Circuit rank test set contains

\(^1\) In restarts, SCIP aborts the current search after encountering sufficient variable bound reductions and starts preprocessing the problem again. For more details about restarts in SCIP, we refer to Section 10.9 in \([1]\).

\(^2\) We used the following SCIP settings: limits/totalnodes = 1, separation/emphasis/aggressive = True, limits/restarts = 0 as well as heuristics/emphasis = Off.

\(^3\) General Algebraic Modeling System (GAMS) Model Library https://www.gams.com/modlib/modlib.htm.
Table 1 Test sets with network and demand loads used in the computational study

| Test set    | Network                  | Total demand |
|-------------|--------------------------|---------------|
|             |                          | 50 | 100 | 200 | 500 | 1000 |
| Belgium     | Belgium                  | ✓  | ✓  | ✓   | ✓   |
| GasLib-40   | GasLib-40                | ✓  | ✓  | ✓   | ✓   |
| Circuit rank| Belgium + 2, 4, 6, 8, 10 arcs | ✓ |

The total network demand load is denoted in \([10^6 \text{ m}^3/\text{day}]\).

instances where the circuit rank is successively increased by adding up to ten new pipelines to the Belgium network, resulting in five new networks Belgium + (2, 4, 6, 8, 10) arcs. The rationale is to add arcs that connect different regions of the network and have a high impact on the network topology. Details about the different networks, their visualization and information of how to generate the new pipelines are found in Table 1, Figure 6 and Section 6 in [21]. We note that the original Belgium and GasLib-40 networks contain compressor stations. However, their treatment is outside the scope of this paper. In the case of the Belgium network, we modeled them as normal pipelines by using available data from the GAMS model library, while in the case of the GasLib-40 network, compressor station arcs had to be contracted by combining their adjacent nodes due to lack of available data.

Demands It is known in practice that increasing network demand typically results in more complex transport situations that stress the networks. Therefore, we generated instances for the Belgium and GasLib-40 test sets that represent a wide range of possible network demand vectors \(b \in \mathbb{R}^{|V|}\) starting from “easier” instances with lower demands up to “harder” instances with higher demands. For the Circuit rank experiment, we utilize demand scenarios with a constant overall network demand. An overview of the total network demand loads (in \([10^6 \text{ m}^3/\text{day}]\)) used for the particular networks is given in Table 1.

Throughout, we randomly generated 500 scenarios for each network and for each given total network demand according to a random uniform distribution of the demand at sources and sinks. For more detailed information about the deployed method, we refer to [21]. Due to the variety of instances, we can assume that the results will provide us with a representative picture for the impact of the convex envelope on gas network related instances.

Expansion capacities In all instances of the three test sets, the expansion capacities \([y_a, \bar{y}_a]\) of all arcs \(a \in A\) result from building at most three pipelines in parallel to the already existing one. The data points for the cost function used follow a quadratic form, see Figure 1 in [21], and were provided to us by practitioners.

Implementation We extended a development version\(^4\) of SCIP by a so-called nonlinear handler, where the separation is applied in the separation and enforcement callbacks. With this handler, we add the cuts (21) and (22) in addition to the cuts that SCIP generates per default (see Sect. 5).

\(^4\) To be released in SCIP 8.
Given that the considered gas instances tend to be numerically unstable due to variables in different scales, we observed that many invalid cuts were generated in some preliminary experiments. The following two measures prevented the addition of invalid cuts to SCIP.

- We only add the cuts (21) and (22) to SCIP if they are violated at least by $10^{-4}$.
- If the LP solution $(\hat{x}, \hat{y})$ satisfies $\hat{y} = y$ or $\hat{y} = \bar{y}$, then we compute a slightly weakened gradient cut by considering the envelopes over the enlarged domain $[x, \bar{x}] \times [y - s, \bar{y}]$ or $[x, \bar{x}] \times [y, \bar{y} + s]$, respectively. We take $s = 10^{-4}$.

Moreover, Propositions 5.1 and 5.2 give conditions under which $\phi_S(x, y) = \text{vex}_D[f](x, y)$. In those cases, we omit the generation of gradient cuts of $\text{vex}_D[f](x, y)$ because they will be generated in any case by SCIP. Specifically,

- If the variable bound in the current LP relaxation satisfies $\bar{x} < \beta \bar{x}$, then we do not generate gradient cuts for $\text{vex}_D[f](x, y)$ (Proposition 5.1).
- If the current LP solution satisfies $T_{x, y}(x) < \beta \bar{x}$ and $\beta \bar{x} < \bar{x}$, i.e., $(x, y) \in A$, then we do not generate gradient cuts for $\text{vex}_D[f](x, y)$ (Proposition 5.2).

**Hardware and software**

The experiments were conducted on a cluster of 64-bit Intel Xeon CPU E5-2670 v2 CPUs at 2.5 GHz with 25 MB cache and 128 GB main memory. To safeguard against a potential mutual slowdown of parallel processes, we bind the processes to specific cores and run at most one job per node at a time. We used a development version of SCIP 6.0.2.4 [10] with CPLEX 12.7.1.0 as LP solver [8] and Ipopt 3.12.13 as NLP solver [36].

### 6.3 Computational Results

In this section, we present the results for the **ROOTGAP** and **TREE** experiments. In the following, **Enabled** refers to adding the gradient cuts of the convex and concave envelopes, as opposed to **Disabled**, which refers to SCIP default settings.

**ROOTGAP Experiment**

From the 6500 instances, we do not consider the ones that have been detected infeasible (396 instances), no primal solution is known (1113 instances), or none of the versions could increase the dual bound by more than $10^{-6}$ (1847 instances). This restriction results in 3144 instances for the **ROOTGAP** experiment.

To compare the dual bounds of **Enabled** and **Disabled** relative to a given primal bound, we use the following measure. For an instance, let $d_1 \in \mathbb{R}$ be the dual bound of **Enabled**, and let $d_2 \in \mathbb{R}$ be the dual bound of **Disabled**. Furthermore, let $p \in \mathbb{R}$ be a reference primal bound, for example, the optimal or best known objective value of that instance. The function $GC : \mathbb{R}^3 \to [-1, 1]$ defined as

$$
GC(d_1, d_2, p) := \begin{cases}
0 & \text{if } d_1 = d_2 \\
+1 - \frac{p - d_1}{p - d_2} & \text{if } d_1 > d_2 \\
-1 + \frac{p - d_2}{p - d_1} & \text{if } d_1 < d_2
\end{cases}
$$

measures the improvement of the gap closed when comparing the dual bounds $d_1$ and $d_2$ relative to $p$, see also [26].
Aggregated results for the gap closed of the ROOTGAP experiment are shown in Table 2 and Fig. 8. Table 2 shows that the addition of the cuts closes more gap on 1033 out of the 3144 instances and closes less gap only on 86 instances. (For an explanation, see below.)

Note that a positive value $GC(d_1, d_2, p) > 0$ implies that the dual bound improved by adding the gradient cuts to the convex envelope $\text{vex}_D[f](x, y)$. Analogously, a negative value $GC(d_1, d_2, p) < 0$ indicates that the addition of the gradient cuts deteriorates the performance.

Aggregated results for the ROOTGAP experiment are shown in Table 2 and Fig. 8. Table 2 shows that the addition of the cuts closes more gap on 1033 out of the 3144 instances and closes less gap only on 86 instances. (For an explanation, see below.)
Additionally, the table presents the average gap closed for instances, where either Enabled improves (better) or deteriorates (worse) by more than 0%, 2%, or 10%, whereas change encompasses the union of those instances that are better or worse by more than 0%, 2%, or 10%.

Among all instances, the average gap closed improvement is 0.87%. However, when one considers all instances for which the gap closed is different, the gap closed improvement increases to 2.44%. Furthermore, if we consider instances for which the gap closed differs by 2% and 10%, then the gap closed improvement increases to 6.75% and 19.69%, respectively.

The presented results show that our convexification of the considered constraint function has a significant impact on the quality of the relaxation. In theory, the dual bounds obtained by Enabled should always be at least as good as the one from Disabled, since we apply the gradient cuts on top of SCIP default. Nevertheless, as mentioned above, there are 86 where disabling the cuts yields improved dual bounds. The explanation for this behavior is that some of the gradient cuts from the convex envelope increase the computational cost of solving LPs in the OBBT propagator [11]. As a consequence, the internal limit of simplex iterations in OBBT are hit faster, which has the side effect that less genvbounds can be learned. However, less genvbounds reduce the number of variable bound reductions, and as a consequence, lead to worse dual bounds. Indeed, when turning off the propagator genvbounds for Enabled and Disabled on these 86 instances, then better dual bounds are obtained by Enabled for all, but 14 out of these 86 instances. Furthermore, if we disable all propagation methods, then the behavior of Enabled is at least as good as Disabled on these 14 instances.5

TREE Experiment Aggregated results for the TREE experiment are shown in Tables 3, 4 and 5 for the three test sets under consideration. We compare the number of solved instances and the solver performance when activating the cut generation for the convex envelope (Enabled) with SCIP using default settings (Disabled). The number of solved instances and the computation time is split into different categories: ALL, ALL OPT, [1, tlim], [10, tlim], [100, tlim] and [1000, tlim]. ALL consists of all instances. ALL OPT consists of the instances that are solved to optimality or detected to be infeasible by both settings. [t, tlim] consists of the instances that could be solved by at least one of both settings, Enabled or Disabled, in more than t seconds within the time limit of one hour. Note that the instances in [t1, tlim] are contained in [t2, tlim] whenever t1 ≥ t2 and that one can consider the instances in [t, tlim] to be harder to solve the larger t is.

Most importantly, on all test sets, enabling the cut generation for the convex envelope increases the number of solved instances and decreases the average solving time compared to SCIP default. Concerning the number of solved instances, Enabled solves 39 more instances on Belgium, 19 more instances on GasLib-40 and 68 more instances on Circuit rank. From the tables, we deduce that all the extra instances that Enabled solves belong to the groups [1000, tlim]. This means that our separation routine renders these instances solvable but not trivially solvable.

5 Furthermore, we did an experiment where we additionally turn off propagation and we saw that Enabled was never worse than Disabled. For more details, see [20].
Table 3 Aggregated results for the TREE experiment on Belgium

|       | # Instances | Enabled # Solved | Disabled # Solved | Time Relative | B&B nodes |
|-------|-------------|-----------------|------------------|--------------|-----------|
| ALL   | 2000        | 1905            | 1866             | 1.19         | –         |
| ALL OPT | 1832        | 1832            | 1832             | 1.19         | 1.22      |
| [1, tlim] | 1635        | 1601            | 1562             | 1.23         | –         |
| [10, tlim] | 1039        | 1005            | 966              | 1.34         | –         |
| [100, tlim] | 542         | 508             | 469              | 1.47         | –         |
| [1000, tlim] | 237         | 203             | 164              | 1.58         | –         |

The column “Relative” reports the change of solving time and branch-and-bound nodes relative to Enabled.

Table 4 Aggregated results for the TREE experiment on GasLib-40

|       | # Instances | Enabled # Solved | Disabled # Solved | Time Relative | B&B nodes |
|-------|-------------|-----------------|------------------|--------------|-----------|
| ALL   | 2000        | 1181            | 1162             | 1.05         | –         |
| ALL OPT | 1108        | 1108            | 1108             | 1.12         | 1.25      |
| [1, tlim] | 1146        | 1092            | 1073             | 1.14         | –         |
| [10, tlim] | 1083        | 1029            | 1010             | 1.15         | –         |
| [100, tlim] | 719         | 665             | 646              | 1.21         | –         |
| [1000, tlim] | 293         | 239             | 220              | 1.20         | –         |

The column “Relative” reports the change of solving time and branch-and-bound nodes relative to Enabled.

Table 5 Aggregated results for the TREE experiment on Circuit rank

|       | # Instances | Enabled # Solved | Disabled # Solved | Time Relative | B&B nodes |
|-------|-------------|-----------------|------------------|--------------|-----------|
| ALL   | 2500        | 1041            | 973              | 1.08         | –         |
| ALL OPT | 766         | 766             | 766              | 1.12         | 1.14      |
| [1, tlim] | 1248        | 1041            | 973              | 1.16         | –         |
| [10, tlim] | 1248        | 1041            | 973              | 1.16         | –         |
| [100, tlim] | 1096        | 889             | 821              | 1.18         | –         |
| [1000, tlim] | 831         | 624             | 556              | 1.16         | –         |

The column “Relative” reports the change of solving time and branch-and-bound nodes relative to Enabled.

To compare the total running time between Enabled and Disabled, we use the shifted geometric mean (sgm) with a shift value of $s = 10$ for the average solving time in seconds. For instances that are solved to optimality by both settings (ALL OPT), we additionally present the sgm for the number of explored branch-and-bound nodes with a shift value of $s = 100$. As can be seen in Tables 3, 4 and 5, there is an
improvement in the performance when using Enabled over SCIP default in all test sets and through all time categories. This improvement increases toward the more “difficult” instances to a speed-up of up to 58% on Belgium. Even for ALL OPT, there is throughout a speed-up of 19% (Belgium) and 12% (GasLib-40 and Circuit rank) when using the Enabled separation routine. Likewise, the number of branch-and-bound nodes significantly decreases by 22%, 25% and 14% for ALL OPT on these three test sets.

Finally, we remark that the Wilcoxon signed-rank test [39] judges the performance improvements on the Belgium, GasLib-40 and Circuit rank test sets as highly statistical significant being consistently distributed in the shifted geometric mean across all three test sets.6

7 Conclusion

In this paper, we derived the convex envelope of the nonconvex and bivariate function \( f(x, y) = y \operatorname{sgn}(x) |x|^\alpha \) for \( \alpha > 1 \). This contribution was motivated by the fact that strong relaxations of nonconvex constraints play a key component in solving MINLPs, see [12]. As this function frequently occurs in expansion planning problems, it is also of significant practical relevance. We performed an extensive computational study on real-world instances which showed the benefit of the convex envelope over the standard relaxation used in state-of-the-art solvers. Given that we provide a closed-form expression for the envelopes, our implementation in the MINLP solver SCIP allows us to calculate and exploit the envelopes at every node in the branch-and-bound tree, which has a crucial impact on the solving process. Apart from yielding improved dual bounds that reduce the gap already in the root node, our procedure drastically boosts the solving process in the branch-and-bound tree. Our procedure significantly reduces the solving time on all test sets, for example, by up to 58% for hard instances on Belgium. Even more important, this tighter relaxation enables to solve more instances on all test sets.

One possible avenue for further research is to consider the convexification of multiple constraints simultaneously. In this fashion, [22] already reported promising results for quadratic functions with absolute value terms for gas network related problems.

Acknowledgements This work was supported by the Research Campus MODAL (Mathematical Optimization and Data Analysis Laboratories) funded by the German Federal Ministry of Education and Research (fund number 05M14ZAM).

Funding Open Access funding enabled and organized by Projekt DEAL.

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

---

6 In this paper, we test against the null hypothesis, which corresponds to the assumption that the approach yielding a better performance is actually not better. For the significance testing, we use a \( p \)-value of 1%, which describes the (approximate) probability that the null hypothesis holds. Small \( p \)-values indicate that the hypothesis was wrong, and we may assert that the difference of the performance values in the shifted geometric mean is indeed significant.
and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Achterberg, T.: Constraint integer programming. Ph.D. thesis, Technische Universität Berlin (2007)
2. Al-Khayyal, F.A., Falk, J.E.: Jointly constrained biconvex programming. Math. Oper. Res. 8(2), 273–286 (1983)
3. Alperovits, E., Shamir, U.: Design of optimal water distribution systems. Water Resour. Res. 13(6), 885–900 (1977). https://doi.org/10.1029/WR013i006p00885
4. André, J., Bonnans, F., Cornibert, L.: Optimization of capacity expansion planning for gas transportation networks. Eur. J. Oper. Res. 197(3), 1019–1027 (2009). https://doi.org/10.1016/j.ejor.2007.12.045
5. Ballerstein, M., Michaels, D., Vigerske, S.: Linear underestimators for bivariate functions with a fixed convexity behavior. Tech. Rep. 13-02, ZIB, Takustr. 7, 14195 Berlin (2013). https://nbn-resolving.de/urn:nbn:de:0297-zib-17641
6. Borraz-Sánchez, C., Bent, R., Backhaus, S., Hijazi, H., Hentenryck, P.V.: Convex relaxations for gas expansion planning. INFORMS J. Comput. 28(4), 645–656 (2016). https://doi.org/10.1287/ijoc.2016.0697
7. Bragalli, C., D’Ambrosio, C., Lee, J., Lodi, A., Toth, P.: On the optimal design of water distribution networks: a practical MINLP approach. Optim. Eng. 13(2), 219–246 (2012). https://doi.org/10.1007/s11081-011-9141-7
8. Cplex, IBM ILOG: CPLEX: High-performance software for mathematical programming and optimization. https://www.ibm.com/analytics/cplex-optimizer (2019). Accessed December 2019
9. Fujiwara, O., Khang, D.B.: A two-phase decomposition method for optimal design of looped water distribution networks. Water Resour. Res. 26(4), 539–549 (1990). https://doi.org/10.1029/WR026i004p00539
10. Gleixner, A., Bastubbe, M., Eifler, L., Gally, T., Gamrath, G., Gottwald, R.L., Hendel, G., Hojný, C., Koch, T., Lübbecke, M.E., Maher, S.J., Miltenberger, M., Müller, B., Pfetsch, M.E., Puchert, C., Rehfeldt, D., Schlösser, F., Schubert, C., Serrano, F., Shinano, Y., Viernickel, J.M., Walter, M., Wegscheider, F., Witt, J.T., Witzig, J.: The SCIP Optimization Suite 6.0, ZIB-Report 18-26, Zuse Institute Berlin (2018). https://nbn-resolving.de/urn:nbn:de:0297-zib-69361
11. Gleixner, A.M., Weltge, S.: Learning and propagating Lagrangian variable bounds for mixed-integer nonlinear programming. In: Gomes, C., Sellmann, M. (eds.) International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Lecture Notes in Computer Science, vol. 7874, pp. 355–361. Springer, Berlin (2013)
12. Grossmann, I.E., Kravanja, Z.: Mixed-integer nonlinear programming: a survey of algorithms and applications. In: Biegler, L., Coleman, T., Conn, A., Santosa, F. (eds.) Large-scale Optimization with Applications, The IMA Volumes in Mathematics and its Applications, vol. 93, pp. 73–100. Springer, New York (1997)
13. Hijazi, H.: Perspective envelopes for bilinear functions. In: AIP Conference Proceedings, vol. 2070. AIP Publishing (2019)
14. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. Springer, Berlin (1990)
15. Humphola, J.: Gas network optimization by MINLP. Ph.D. thesis, Technische Universität Berlin (2014). https://doi.org/10.14279/depositonce-4255
16. Jach, M., Michaels, D., Weismantel, R.: The convex envelope of \((n - 1)\)-convex functions. SIAM J. Optim. 19(3), 1451–1466 (2008)
17. Jacoby, S.L.: Design of optimal hydraulic networks. J. Hydraul. Div. 94(3), 641–662 (1968)
18. Karmeli, D., Gadish, Y., Meyers, S.: Design of optimal water distribution networks. J. Pipeline Div. 94(1), 1–10 (1968)
19. Katz, D.L.V.: Handbook of Natural Gas Engineering. McGraw-Hill, New York (1959)
20. Lenz, R.: Optimization of stationary expansion planning and transient network control by mixed-integer nonlinear programming. Ph.D. thesis, Technische Universität Berlin (2021). https://doi.org/10.14279/depositonce-12765

21. Lenz, R., Becker, K.H.: Optimization of capacity expansion in potential-driven networks including multiple looping: a comparison of modelling approaches. OR Spectrum 44(1), 179–224 (2021). https://doi.org/10.1007/s00291-021-00648-7

22. Liers, F., Martin, A., Merkert, M., Mertens, N., Michaels, D.: Solving mixed-integer nonlinear optimization problems using simultaneous convexification: a case study for gas networks. J. Global Optim. 80(2), 1–34 (2021). https://doi.org/10.1007/s10898-020-00974-0

23. Linderoth, J.: A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. Math. Program. 103(2), 251–282 (2005). https://doi.org/10.1007/s10107-005-0582-7

24. Locatelli, M.: Convex envelopes of bivariate functions through the solution of KKT systems. J. Global Optim. 72(2), 277–303 (2018). https://doi.org/10.1007/s10898-018-0626-1

25. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part 1—convex underestimating problems. Math. Program. 10(1), 147–175 (1976)

26. Müller, B., Muñoz, G., Gasse, M., Gleixner, A., Lodi, A., Serrano, F.: On generalized surrogate duality in mixed-integer nonlinear programming. Math. Program. (2021). https://doi.org/10.1007/s10107-021-01691-6

27. Raghunathan, A.U.: Global optimization of nonlinear network design. SIAM J. Optim. 23(1), 268–295 (2013). https://doi.org/10.1137/110827387

28. Rockafellar, R.T.: Convex Analysis. Princeton Mathematical Series, vol. 28. Princeton University Press, Princeton (1970)

29. Schmidt, M., Allmann, D., Burlacu, R., Humbola, J., Joormann, I., Kanelakis, N., Koch, T., Oucherif, D., Pletsch, M.E., Schewe, L., Schwarz, R., Sirvent, M.: Gaslib—a library of gas network instances. Data 2(4) (2017). https://doi.org/10.3390/data2040040

30. Sherali, H.D., Alameddine, A.: An explicit characterization of the convex envelope of a bivariate bilinear function over special polytopes. Ann. Oper. Res. 25(1), 197–209 (1990)

31. Shiono, N., Suzuki, H.: Optimal pipe-sizing problem of tree-shaped gas distribution networks. Eur. J. Oper. Res. 252(2), 550–560 (2016). https://doi.org/10.1016/j.ejor.2016.01.008

32. Smith, E.M., Pantelides, C.C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. Comput. Chem. Eng. 23(4–5), 457–478 (1999)

33. Tawarmalani, M., Sahinidis, N.V.: Semidefinite relaxations of fractional programs via novel convexification techniques. J. Global Optim. 20, 137–158 (2001)

34. Tawarmalani, M., Sahinidis, N.V.: Convex extensions and convex envelopes of lower semi-continuous functions. Math. Program. 93, 247–263 (2002)

35. Vigerske, S.: Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming. Ph.D. thesis, Humboldt-Universität zu Berlin (2012)

36. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106(1), 25–57 (2006)

37. Walski, T.M., Chase, D.V., Savic, D.A.: Water distribution modeling. Tech. rep., University of Dayton, Civil and Environmental Engineering and Engineering Mechanics Faculty Publications, Paper 17 (2001)

38. Weymouth, T.R.: Problems in natural gas engineering. Trans. Am. Soc. Mech. Eng. 34(1349), 185–231 (1912)

39. Wilcoxon, F.: Individual comparisons by ranking methods. In: Kotz, S., Johnson, N.L. (eds.) Breakthroughs in Statistics, pp. 196–202. Springer, Berlin (1992)

40. Zamora, J.M., Grossmann, I.E.: A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. J. Global Optim. 14(3), 217–249 (1999)

41. Zhang, J., Zhu, D.: A bilevel programming method for pipe network optimization. SIAM J. Optim. 6(3), 838–857 (1996). https://doi.org/10.1137/S1052623493260696

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