Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

RALF LENZ AND FELIPE SERRANO

Tight Convex Relaxations for the Expansion Planning Problem

ZIB Report 21-05 (January 2021)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Tight Convex Relaxations for the Expansion Planning Problem

Ralf Lenz *and Felipe Serrano †

February 9, 2021

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 improve 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%.

1 Introduction

The goal of this paper is to improve the algorithmic performance of the MINLP solver for the expansion planning problem of energy networks. Given the increasing demands of modern society, these type of problems are becoming more relevant as well as more challenging. For more details on the problem we consider see Section 2.

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 $\{x \in D \,|\, f(x) \leq 0\}$, where $D \subseteq \mathbb{R}^n$ is convex, the most common method for constructing a convex relaxation is to compute a *convex*

*Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, lenz@zib.de
†Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, serrano@zib.de

underestimator, that is, a function $g\colon D \to \mathbb{R}$ 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 $\mathrm{vex}_D(f)$. 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 $f(x, y) = y\,\mathrm{sgn}(x)|x|^\alpha$ with $\alpha > 1$. This is the only source of nonlinearity and nonconvexity of the expansion planning problems that we consider, see Section 2. Thus, the convex relaxations depend on the convexification of these constraints. As we will see in Section 4.1, the concave envelope of f can be deduced from the convex envelope of f, 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.

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% percent for hard instances.

Outline The rest of the paper is structured as follows: In Section 3, we introduce preliminary theoretical results from convex analysis that we need to derive the convex envelope of $y\,\mathrm{sgn}(x)|x|^\alpha$. In Section 4, we present our main contribution, which is the derivation of a closed-form expression for the convex envelope of f. In Section 5, we compare the convex envelope with the standard factorable relaxation for the constraint function f followed by an extensive computational study in Section 6. We test the impact of the convex envelope over real-world expansion planning problem instances using the MINLP solve 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, Section 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 as, 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 following, we state a model formulation for the loop expansion planning problem. 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 in 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 $x_a \in [\underline{x}_a, \overline{x}_a]$ for all arcs $a = (v, w) \in \mathcal{A}$ and non-negative potential variables $\pi_v \in [\underline{\pi}_v, \overline{\pi}_v]$ at each node $v \in \mathcal{V}$. The arc flows arise from a friction-induced potential difference between the arcs' adjacent nodes modeled by Equation (1a). Note that $\operatorname{sgn}(x_a)$ indicates the direction of the arc flow. Given that, in general, bi-directional arc flow is possible, we assume that $\underline{x} < 0$ and $\overline{x} > 0$. Here, for each pipe $a \in \mathcal{A}$, positive variables $y_a \in [\underline{y}_a, \overline{y}_a]$ model the physical impact of the expansion on the potential loss along the arc. The variables $c_a \geq 0$ represent the corresponding costs of pipe a that can be assumed as proportional to the pipe length L_a. We recall from [1], that the setup of the y_a and c_a variables enables to split the entire length of the pipeline to be looped into several segments of variable lengths each of which may have its own diameter from a discrete set, called split-pipe approach. Note that in the case that pipe a is not looped, then by construction $y_a = \underline{y}_a$ and the associated costs $c_a = 0$. Otherwise, $y_a > \underline{y}$ and $c_a > 0$. [1] shows that in the split-pipe setting, the best expansion candidates are determined by the efficient frontier (Equation (1b)), which is given by k_a-many non-dominated points in the (y_a, c_a)-system. Finally, as in classical network flow problems, Equation (1c) models the flow conservation at every network node. Then, the model reads,

$$
\begin{aligned}
\min_{y,c,x,\pi} \quad & \sum\nolimits_{a\in\mathcal{A}} L_a c_a \\
\text{subject to} \quad & \pi_v - \pi_w = y_a \operatorname{sgn}(x_a)|x_a|^\alpha && \forall a = (v,w) \in \mathcal{A}, && \text{(1a)}\\
& c_a \geq s_{a,i} y_a + t_{a,i} && \forall a \in A\ \forall i \in [k_a - 1], && \text{(1b)}\\
& \sum_{a\in\delta^+(v)} x_a - \sum_{a\in\delta^-(v)} x_a = b_v && \forall\, v \in \mathcal{V}, && \text{(1c)}\\
& \underline{\pi}_v \leq \pi_v \leq \overline{\pi}_v && \forall\, v \in \mathcal{V},\\
& \underline{x}_a \leq x_a \leq \overline{x}_a && \forall\, a \in \mathcal{A},\\
& \underline{y}_a \leq y_a \leq \overline{y}_a && \forall\, a \in \mathcal{A},\\
& c_a \geq 0 && \forall\, a \in \mathcal{V}.
\end{aligned}
$$

In this paper, we compute the convex hull of the constraint (1a), by studying the convex and concave envelope of the nonlinear function $y\,\mathrm{sgn}(x)|x|^\alpha$.

Finally, let us mention that, in general, these type of optimization problems get more challenging to solve on networks that have a higher *circuit rank*, which is the number $|\mathcal{A}| - |\mathcal{V}| + 1$, because the existence of cycles leads to more complex patterns of flow directions, see [2] and the references therein.

3 Introduction to Convex Envelopes

In this section, we first introduce some mathematical notation and necessary theoretical results that are required to derive $\mathrm{vex}_D(f)$. Afterwards, we give a brief overview of related literature on convex envelopes. We only consider lower semicontinuous functions, which we define as follows.

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

$$epi_D f = \{(x,\mu)|\ x \in D, \mu \in \mathbb{R}, \mu \geq f(x)\}.$$

Definition 3.2. *A function $f\colon D \to \mathbb{R}$ 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 $f\colon D \to \mathbb{R}$ over a convex set $D \subseteq \mathbb{R}^n$ is given by the tightest convex underestimator $\eta\colon D \to \mathbb{R}$ of f, defined pointwise as*

$$vex_D[f](x) = \sup\{\ \eta(x) \mid \eta(y) \leq f(y) \text{ for all } y \in D, \eta \text{ convex}\ \}.$$

The concave envelope *of a function $f\colon D \to \mathbb{R}$ over $D \subseteq \mathbb{R}^n$ is given by the tightest concave overestimator $\eta\colon D \to \mathbb{R}$ of f, defined pointwise as*

$$cave_D[f](x) = \inf\{\ \eta(x) \mid \eta(y) \leq f(y) \text{ for all } y \in D, \eta \text{ concave}\ \}.$$

A geometrical characterization of the convex envelope is as follows.

Theorem 3.4 ([3], Theorem 5.3)**.** *Let E be any convex set in $\mathbb{R}^{n+1}$ and let*

$$g(x) = \inf\{\mu|\ (x,\mu) \in E\}.$$

Then $g\colon \mathbb{R}^n \to \mathbb{R}$ is a convex function.

The convex envelope of a l.s.c. function f corresponds to the function g, obtained by applying Theorem 3.4 to the convex set $\mathrm{conv}(\mathrm{epi}_D f)$. Therefore, it follows

$$\mathrm{vex}_D[f](x) = \min\{\mu|\ (x,\mu) \in \mathrm{conv}(\mathrm{epi}_D f)\}, \tag{2}$$

see also [3]. 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

$$\begin{aligned} \operatorname{vex}_D[f](x) = \min & \sum_{k=1}^{n+1} \lambda_k \, f(x_k) \\ & \sum_{k=1}^{n+1} \lambda_k \, x_k = x \\ & \sum_{k=1}^{n+1} \lambda_k = 1 \\ \lambda_k \geq 0, x_k \in D \ \text{ for } \ & 1 \leq k \leq n+1. \end{aligned} \tag{3}$$

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

Definition 3.5. *Let D be compact and convex subset of $\mathbb{R}^n$ and let $f\colon D \to \mathbb{R}$ be a l.s.c. function. The* generating set *of the convex envelope of f is defined by*

$$\mathcal{G}_D(f) := \{x \in D \mid (x, vex_D[f](x)) \text{ is an extreme point of } conv\,(epi_D f)\}.$$

Given that a closed convex set with no lines is characterized by the set of its extreme points and extreme directions ([3, Theorem 18.5]), we have the following result, see also [4].

Theorem 3.6. *Let D be compact and convex subset of $\mathbb{R}^n$, $f\colon D \to \mathbb{R}$ be a l.s.c. function, and $A \subseteq D$. Then $conv(epi_D f) = conv(epi_A f)$ if and only if $\mathcal{G}_D(f) \subseteq A$.*

Even though there exists no general formula to compute $\mathcal{G}_D(f)$ of a nonconvex function f, [4] provide a condition that determines when a point $x \in D$ is not in $\mathcal{G}_D(f)$. For this we need the relative interior of a convex set D, $\operatorname{ri}(D) := \{x \in D : \forall y \in D \; \exists \lambda < 1 : \lambda x + (1-\lambda) y \in D\}$.

Corollary 3.7 ([4], Corollary 7)**.** *Let $f\colon D \to \mathbb{R}$ 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 $x \in \operatorname{ri}(l)$ and f restricted to l is concave, then $x \notin \mathcal{G}_D(f)$.*

Literature review. 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, [5]. It can be shown that for a l.s.c. function f over a compact and convex domain D, $\min_{x \in D} f(x) = \min_{x \in D} \operatorname{vex}_D[f](x)$ and hence computing the convex envelope

is in general $\mathcal{NP}$-hard. Unless $\mathcal{P} = \mathcal{NP}$, 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, [6] constructs a linear relaxation over a box domain for the bilinear term $x\,y$, which is proven to be the convex envelope by [7]. In the last decade, several approaches arose to convexify the bilinear term $x\,y$ over special domains, see, e.g., [8], [9], [10], [11], and [12]. The convex envelope of the fractional term y/x has been studied, e.g., in [13], and [14].

Moreover, [14] computes the convex envelope of the function $f(x, y) = g(y)\,x^2$ over a rectangular domain $D \subseteq \mathbb{R}^2$, where $g\colon D \to \mathbb{R}$ is convex in x and concave in y. Even though this function is close to our constraint function $f(x, y) = y\,\mathrm{sgn}(x)|x|^\alpha$, this approach is not applicable in our setting, given that our function involves the nonconvex term $\mathrm{sgn}(x)|x|^\alpha$ instead of x^2.

Note that $f(x, y) = y\,\mathrm{sgn}(x)|x|^\alpha$ is twice differentiable only for $\alpha > 2$ and thus it is not possible to use techniques based on the Hessian, as done, for example, for $(n-1)$-convex functions in [15]. 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 [16] for twice continuously differentiable functions with fixed convexity behavior over a box. However, our analysis is valid for any $\alpha > 1$. Furthermore, we note that for relevant applications the values of α are $\alpha = 2$ for gas network operations [17] and $\alpha = 1.852$ for water network operations, see [18].

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

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

4.1 Preliminaries about the envelopes of $f(x, y) = y\,sgn(x)|x|^\alpha$

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

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

$$\sum_{k=1}^{3} \lambda_k \, y_k = y,$$
$$\sum_{k=1}^{3} \lambda_k = 1,$$
$$\lambda_k \geq 0, (x_k, y_k) \in D \quad \forall 1 \leq k \leq 3.$$

Note that the above problem has nine variables (x_k, y_k, λ_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 $f(x, y) = y \operatorname{sgn}(x)|x|^\alpha$ allows us to deduce both envelopes from each other, see Figure 1. The function $f(x, y) = y \operatorname{sgn}(x)|x|^\alpha$ 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.

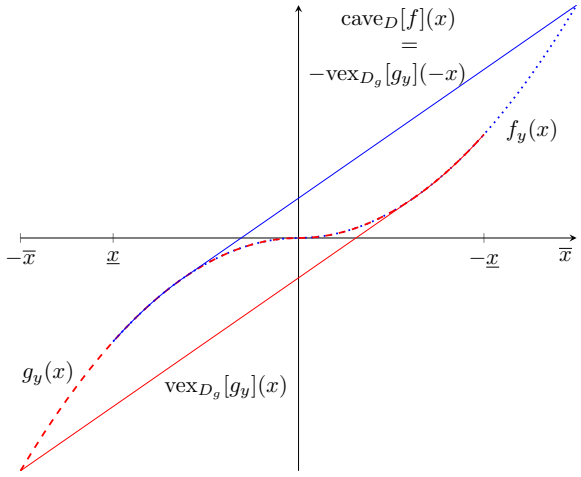

Figure 1: Illustration of Proposition 4.1 for the one-dimensional functions $f_y(x) = y \operatorname{sgn}(x) x^2$ (blue, dotted) and $g_y(x) := -f_y(-x) = -y \operatorname{sgn}(-x)(-x)^2$ (red, dashed) with $f_y \colon D := [\underline{x}, \overline{x}] \to \mathbb{R}$ and $g_y \colon D_g := [-\overline{x}, -\underline{x}] \to \mathbb{R}$ for fixed y, where $\operatorname{vex}_{D_f}[g_y](x)$ (red, solid) is the convex envelope of g_y and $-\operatorname{vex}_{D_g}[g_y](-x) = \operatorname{cave}_D[f](x)$ (blue, solid) the concave envelope of f_y.

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

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

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

Thus,

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

□

4.2 Problem reduction by using the generating set

To simplify Problem (3), we use Corollary 3.7 to find a set $A \subseteq D$ such that $\mathcal{G}_D(f) \subseteq A$ for $f(x,y) = y\,\mathrm{sgn}(x)|x|^\alpha$.

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 3.7 implies that all points in $\mathrm{ri}(l) = \{(x,y) \,|\, y \in (\underline{y}, \overline{y})\}$ are not in $\mathcal{G}_D(f)$. Thus,

$$\mathcal{G}_D(f) \subseteq \{(x,\underline{y}),\, \underline{x} \le x \le \overline{x}\,\} \cup \{(x,\overline{y}),\, \underline{x} \le x \le \overline{x}\,\}.$$

Let $D_{\underline{y}} := \{(x,\underline{y}) \,|\, \underline{x} \le x \le \overline{x}\}$ and $D_{\overline{y}} := \{(x,\overline{y}) \,|\, \underline{x} \le x \le \overline{x}\}$. Given that $\mathcal{G}_D(f) \subset D_{\underline{y}} \cup D_{\overline{y}}$, Theorem 3.6 allows us to express the convex hull of $\mathrm{epi}_D f$ as

$$\begin{aligned} \mathrm{conv}(\mathrm{epi}_D f) &= \mathrm{conv}(\mathrm{epi}_{D_{\underline{y}} \cup D_{\overline{y}}} f) \\ &= \mathrm{conv}(\mathrm{epi}_{D_{\underline{y}}} f \cup \mathrm{epi}_{D_{\overline{y}}} f) \\ &= \mathrm{conv}(\mathrm{conv}(\mathrm{epi}_{D_{\underline{y}}} f) \cup \mathrm{conv}(\mathrm{epi}_{D_{\overline{y}}} f)). \end{aligned} \tag{4}$$

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

Proposition 4.2 ([19], Section 7.5.2)**.** *The convex envelope of the function $f_y\colon [\underline{x}, \overline{x}] \to \mathbb{R}$ given by $f_y(x) = y\,sgn(x)|x|^\alpha$, where $\underline{x} < 0 < \overline{x}$, $\alpha > 1$, and $y > 0$ is given by $\varphi_y\colon [\underline{x}, \overline{x}] \to \mathbb{R}$, with*

$$\varphi_y(x) = \begin{cases} yx^\alpha & x \ge \beta\underline{x} \\ \alpha(\beta\underline{x})^{\alpha-1}xy + (1-\alpha)(\beta\underline{x})^\alpha y & x < \beta\underline{x}, \end{cases} \tag{5}$$

where β is the unique negative root of $(\alpha-1)(-\beta)^\alpha + \alpha(-\beta)^{\alpha-1} - 1$.

Remark 4.3. *The function $(x,y) \mapsto \varphi_y(x)$ is not the convex envelope of $f(x,y)$ as can be seen from Figure 6 (middle). This function is not even convex!* □

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

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

Therefore, we have

$$\mathrm{vex}_D[f](x,y) = \min\left\{\mu |\ (x,y,\mu) \in \mathrm{conv}\left(\mathrm{epi}_{D_{\underline{y}}} \varphi_{\underline{y}} \cup \mathrm{epi}_{D_{\overline{y}}} \varphi_{\overline{y}}\right)\right\}.$$

Equivalently,

$$\mathrm{vex}_D[f](x,y) = \min_{x_1, x_2, \lambda} (1-\lambda)\varphi_{\underline{y}}(x_1) + \lambda\varphi_{\overline{y}}(x_2)$$

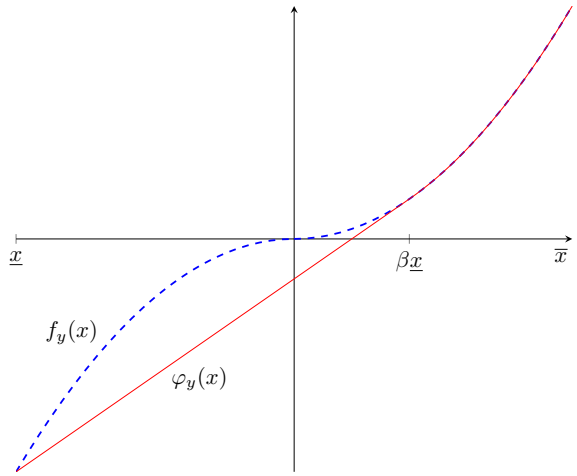

Figure 2: The convex envelope φ_y (red) of the one-dimensional function f_y (blue) depends on the relation of the values $\beta\underline{x}$ and $\overline{x}$. In the case of $\beta\underline{x} < \overline{x}$, the convex envelope is given by the secant between the points $(\underline{x}, f_y(\underline{x}))$ and $(\beta\underline{x}, f_y(\beta\underline{x}))$ for $x \in [\underline{x}, \beta\underline{x}]$ and the function $f_y(x)$ itself for $x \in [\beta\underline{x}, \overline{x}]$. Otherwise, if $\beta\underline{x} > \overline{x}$, then $\mathrm{vex}_{[\underline{x},\overline{x}]}[f_y](x)$ is given by the secant between $(\underline{x}, f_y(\underline{x}))$ and $(\overline{x}, f_y(\overline{x}))$.

$$\begin{aligned} &\text{s.t. } (1-\lambda)x_1 + \lambda x_2 = x && \text{(6a)}\\ &\quad (1-\lambda)\underline{y} + \lambda\overline{y} = y && \text{(6b)}\\ &\quad \lambda \in [0,1], x_1, x_2 \in [\underline{x}, \overline{x}]. \end{aligned}$$

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

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 λ

$$\lambda_y := \lambda = \frac{y - \underline{y}}{\overline{y} - \underline{y}} \in (0, 1).$$

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

$$t_{x,y}(z) = \frac{x - (1-\lambda_y)z}{\lambda_y}, \quad \text{and} \tag{7}$$

$$T_{x,y}(z) = \frac{x - \lambda_y z}{1 - \lambda_y}. \tag{8}$$

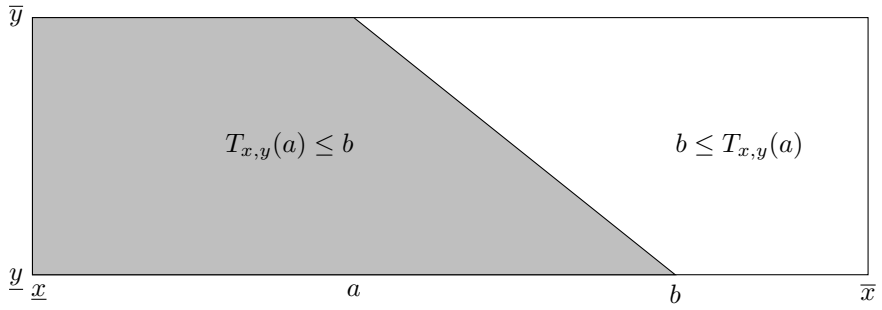

Figure 3: An interpretation of function $T_{x,y}$ on $D = [\underline{x}, \overline{x}] \times (\underline{y}, \overline{y})$. The set $\{(x, y) \colon T_{x,y}(a) \leq b\}$ with $\underline{x} \leq a \leq b \leq \overline{x}$ represents the gray shaded region in D, and $\{(x, y) \colon b \leq T_{x,y}(a)\}$ corresponds to the region in white in D. Both regions are separated by the line segment that connects the points $(a, \overline{y})$ and $(b, \underline{y})$.

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 Figure 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

$$\begin{aligned} \min\ & (1 - \lambda_y)\varphi_{\underline{y}}(x_1) + \lambda_y \varphi_{\overline{y}}(t_{x,y}(x_1)) \\ \text{s.t. } & \max\{\underline{x}, T_{x,y}(\overline{x})\} \leq x_1 \leq \min\{\overline{x}, T_{x,y}(\underline{x})\}. \end{aligned} \tag{9}$$

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

$$F_{x,y}(x_1) = (1 - \lambda_y)\varphi_{\underline{y}}(x_1) + \lambda_y \varphi_{\overline{y}}(t_{x,y}(x_1)). \tag{10}$$

Then, Problem (9) is equivalent to

$$\begin{aligned} \min\ & F_{x,y}(x_1) \\ \text{s.t. } & \underline{x}_1 \leq x_1 \leq \overline{x}_1. \end{aligned} \tag{11}$$

Remark 4.4. *Problem* (11) *allows for a geometrical interpretation as illustrated in Figure 4: For a given point $(x, y) \in D$, Problem* (11) *corresponds to finding the points $(x_1, \underline{y})$ and $(t_{x,y}(x_1), \overline{y})$ such that the line segment between these points contains (x, y), and the line segment with endpoints $(x_1, \underline{y}, \varphi_{\underline{y}}(x_1))$ and $(t_{x,y}(x_1), \overline{y}, \varphi_{\overline{y}}(t_{x,y}(x_1)))$ 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 $x_2 = t_{x,y}(x_1)$.* □

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 $\varphi_{\overline{y}}(t_{x,y}(\cdot))$ is convex since it is the composition of a convex ($\varphi_{\overline{y}}$) with an affine linear function ($t_{x,y}$).

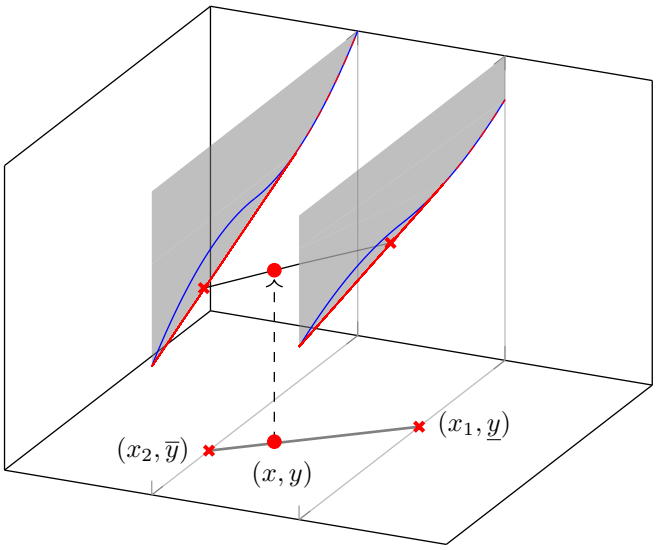

Figure 4: A geometrical interpretation of Problem (11) to determine $\operatorname{vex}_D[f](x, y)$, as described in Remark 4.4. The one-dimensional functions in blue $f_{\underline{y}}(x) = \underline{y}\operatorname{sgn}(x)|x|^\alpha$ and $f_{\overline{y}}(x) = \overline{y}\operatorname{sgn}(x)|x|^\alpha$ are obtained after fixing y to $\underline{y}$ and $\overline{y}$ in $f(x, y) = y\operatorname{sgn}(x)|x|^\alpha$. Their corresponding convex envelopes $\varphi_{\underline{y}}(x)$ and $\varphi_{\overline{y}}(x)$ given by (5) are shown in red, and their epigraphs $\operatorname{epi}_{D_{\underline{y}}}\varphi_{\underline{y}}$ and $\operatorname{epi}_{D_{\overline{y}}}\varphi_{\overline{y}}$ in gray. Red crosses in the (x, y)–space correspond to the points $(\overline{x}_1, \underline{y})$ and $(t_{x,y}(x_1), \overline{y})$ with $x_2 = t_{x,y}(x_1)$, and red crosses in the $(x, y, \varphi_y(x_1))$–space correspond to the points $(x_1, \underline{y}, \varphi_{\underline{y}}(x_1))$ and $(t_{x,y}(x_1), \overline{y}, \varphi_{\overline{y}}(t_{x,y}(x_1)))$.

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

Proposition 4.5. *Suppose that a function $F\colon \mathbb{R} \to \mathbb{R}$ is convex and has a global minimum $z^* \in \mathbb{R}$. Then the solution of $\min\{F(z) : a \le z \le b\}$ is given by*

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

where $\operatorname{mid}(a, b, z^*)$ 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., $a \le z^* \le b$, then $F(\operatorname{mid}(a, b, z^*)) = F(z^*)$. If $z^* \le a$, then the convexity of F implies that F is increasing in $[z^*, \infty)$ and so $F(a) = F(\operatorname{mid}(a, b, z^*))$ is the minimum. The argument is analogous for the case $z^* \ge b$. □

In order to apply Proposition 4.5 we need to extend $F_{x,y}$ from $x_1 \in \underline{x}_1 \le \overline{x}_1$ 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 4.6. *The function* $F_{x,y}\colon \mathbb{R} \to \mathbb{R}$*, given by Equation* (10) *with* $\alpha > 1$*, is coercive, i.e., it satisfies*

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

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

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

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

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

This can be rewritten as $a x_1^{\alpha} + b x_1 + c$, 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, $\lim_{x_1 \to \infty} F_{x,y}(x_1) = \infty$. □

Figure 5 visualizes function $F_{x,y}(x_1)$ for different pairs of points (x, y).

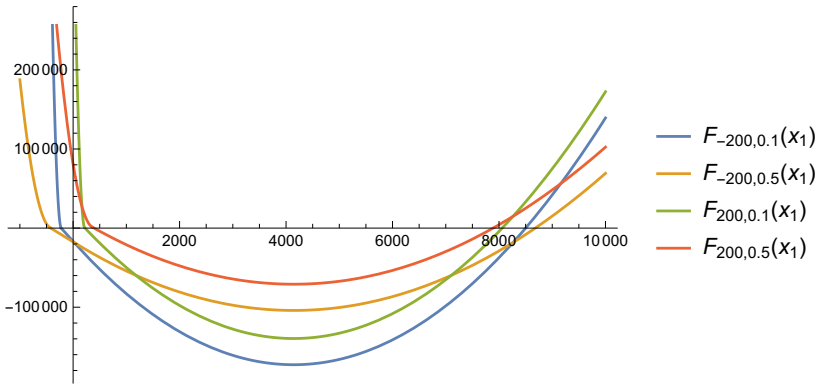

Figure 5: An illustration of function $F_{x,y}(x_1)$ for different pairs of values $(x, y) \in \{-200, 200\} \times \{0.1, 0.5\}$. The function $F_{x,y}$ is continuous, convex and coercive.

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,

$$\begin{aligned} F'_{x,y}(x_1) &= (1 - \lambda_y)\varphi'_{\underline{y}}(x_1) + \lambda_y \varphi'_{\overline{y}}\left(t_{x,y}(x_1)\right) t'_{x,y}(x_1) \\ &\overset{(7)}{=} (1 - \lambda_y)\Big(\varphi'_{\underline{y}}(x_1) - \varphi'_{\overline{y}}(t_{x,y}(x_1)) \Big). \end{aligned}$$

Then, given the piecewise nature of φ_y, see (5), we get

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

where

$$\begin{aligned} LL &= \{x_1 \in \mathbb{R}\colon x_1 \le \beta\underline{x},\ t_{x,y}(x_1) \le \beta\underline{x}\},\\ LR &= \{x_1 \in \mathbb{R}\colon x_1 \le \beta\underline{x},\ t_{x,y}(x_1) \ge \beta\underline{x}\},\\ RL &= \{x_1 \in \mathbb{R}\colon x_1 \ge \beta\underline{x},\ t_{x,y}(x_1) \le \beta\underline{x}\},\\ RR &= \{x_1 \in \mathbb{R}\colon x_1 \ge \beta\underline{x},\ t_{x,y}(x_1) \ge \beta\underline{x}\}. \end{aligned}$$

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

Case $x_1 \in LL$: From (12) follows that $F'_{x,y}(x_1) = 0$ if and only if

$$\alpha(\beta\underline{x})^{\alpha-1}\underline{y} = \alpha(\beta\underline{x})^{\alpha-1}\overline{y}.$$

Given that $\underline{y} < \overline{y}$, 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

$$x_{lr} := T_{x,y}(\beta\underline{x}\,(\underline{y}/\overline{y})^{\frac{1}{\alpha-1}}).$$

Note, however, that $x_{lr} \notin LR$, as $t_{x,y}(x_{lr}) < \beta\underline{x}$. 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

$$x_{rl} := \beta\underline{x}\left(\frac{\overline{y}}{\underline{y}}\right)^{\frac{1}{\alpha-1}}. \tag{13}$$

The point x_{rl} is in RL if and only if $x_{rl} \ge \beta\underline{x}$ and $t_{x,y}(x_{rl}) \le \beta\underline{x}$. Given that $\beta, \underline{x} < 0$, we have that $x_{rl} > \beta\underline{x}$. Thus, the global minimum is x_{rl} if and only if $t_{x,y}(x_{rl}) \le \beta\underline{x}$. 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

$$x_{rr} := x\frac{\overline{y}^{\frac{1}{\alpha-1}}}{\lambda_y\underline{y}^{\frac{1}{\alpha-1}} + (1-\lambda_y)\overline{y}^{\frac{1}{\alpha-1}}}. \tag{14}$$

From the above case distinction, it follows that the optimal solution is x_{rl} if $t_{x,y}(x_{rl}) \leq \beta \underline{x}$, otherwise it is x_{rr}. Therefore, from Proposition 4.5, we conclude that

$$\operatorname{vex}_D[f](x,y) = \begin{cases} F_{x,y}(\operatorname{mid}(\underline{x}_1, \overline{x}_1, x_{rl})) & \text{if } t_{x,y}(x_{rl}) \leq \beta \underline{x} \\ F_{x,y}(\operatorname{mid}(\underline{x}_1, \overline{x}_1, x_{rr})) & \text{else.} \end{cases} \tag{15}$$

Figure 6 shows a visualization of $y \operatorname{sgn}(x)|x|^\alpha$ and its convex envelope.

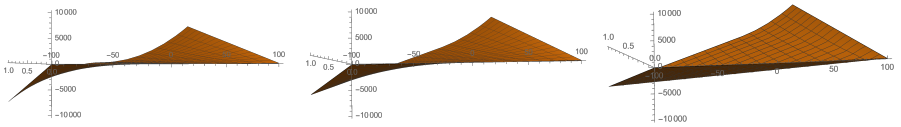

Figure 6: An illustration of the functions f, $(x,y) \mapsto \varphi_y(x)$, and $\operatorname{vex}_D[f](x,y)$ over the domain $(x,y) \in [-100,100] \times [0.01,1]$ for $\alpha = 2$.

4.6 Simplifying the convex envelope.

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

Theorem 4.7. *The convex envelope of $f(x,y) = y\, sgn(x)|x|^\alpha$ over $D = [\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}]$ is given by*

$$vex_D[f](x,y) = F_{x,y}(\min\{\overline{x}, T_{x,y}(\underline{x}), \max\{x_{rl}, x_{rr}\}\}). \tag{16}$$

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

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

where x_{rl} and x_{rr} are given in (13) and (14). Then, we can rewrite (15) as

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

Then, we show that

$$\underline{x}_1 \leq \max\{x_{rl}, x_{rr}\}. \tag{18}$$

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

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

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

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

The equivalence (17) follows from the following computation,

$$x_{rl} \leq x_{rr} \iff G_{x,y}(x_{rl}) \leq G_{x,y}(x_{rr})$$

$$
\begin{aligned}
&\iff G_{x,y}(x_{rl}) \leq 0 \\
&\iff x_{rl} - t_{x,y}(x_{rl}) \left(\frac{\overline{y}}{\underline{y}}\right)^{\frac{1}{\alpha-1}} \leq 0 \\
&\iff x_{rl} \leq t_{x,y}(x_{rl}) \left(\frac{\overline{y}}{\underline{y}}\right)^{\frac{1}{\alpha-1}} \\
&\iff \beta \underline{x} \leq t_{x,y}(x_{rl}).
\end{aligned}
$$

To prove (18), it is enough to show $\underline{x}_1 \leq x_{rr}$. Recall that $\underline{x}_1 = \max\{\underline{x}, T_{x,y}(\overline{x})\}$. The inequality $\underline{x} \leq x_{rr}$ follows from the definition of x_{rr} and $\lambda_y, \underline{y}, \overline{y} > 0$.

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

$$
\begin{aligned}
T_{x,y}(\overline{x}) \leq x_{rr} &\iff G_{x,y}(T_{x,y}(\overline{x})) \leq 0 \\
&\iff T_{x,y}(\overline{x}) \leq \overline{x} \left(\frac{\overline{y}}{\underline{y}}\right)^{\frac{1}{\alpha-1}} \\
&\iff x \leq (1-\lambda_y)\overline{x} \left(\frac{\overline{y}}{\underline{y}}\right)^{\frac{1}{\alpha-1}} + \lambda_y \overline{x},
\end{aligned}
$$

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

Remark 4.8. *From* (16) *we see that if* $x_{rl} \geq \overline{x}$*, then*

$$vex_D[f](x,y) = F_{x,y}(\min\{\overline{x}, T_{x,y}(\underline{x})\}) = F_{x,y}(\overline{x}_1).$$

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

In summary, we derived an analytic solution for the convex envelope of the function $f(x,y) = y\,\mathrm{sgn}(x)|x|^\alpha$, with $\alpha > 1$, which is stated in (16). To evaluate the convex envelope only requires the computation of the function $F_{x,y}$ at the minimum between $\overline{x}_1$, x_{rl}, and x_{rr}.

5 Convexification of $y\,sgn(x)|x|^\alpha$ in SCIP

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

$$\{(x,y,\theta) \in D \times \mathbb{R} : \theta \geq y\,\mathrm{sgn}(x)|x|^\alpha\}. \tag{19}$$

By virtue of Theorem 3.4, any convex relaxation of (19) induces a convex underestimator of $y\,\mathrm{sgn}(x)|x|^\alpha$. Therefore, we construct the convex underestimator of $y\,\mathrm{sgn}(x)|x|^\alpha$ induced by SCIP's convex relaxation of (19) and compare it to the convex envelope of $y\,\mathrm{sgn}(x)|x|^\alpha$.

SCIP, as other MINLP solvers, use 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 [20], and for a description of how it works in SCIP to [19].

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

$$\begin{aligned} z &= \mathrm{sgn}(x)|x|^\alpha, \\ \theta &\geq yz. \end{aligned}$$

Then, SCIP builds a relaxation by computing the convex envelopes of $f_1(x) = \mathrm{sgn}(x)|x|^\alpha$ and $f_2(y,z) = y\,z$. Note that by Proposition 4.2, the convex envelope of f_1 is φ_y with $y = 1$, which we denote by φ_1. Moreover, the convex envelope of f_2 is given by the well-known McCormick inequalities [6],

$$\psi(y,z) := \max\left\{\underline{y}z + y\underline{z} - \underline{y}\underline{z},\; \overline{y}z + y\overline{z} - \overline{y}\,\overline{z}\right\},$$

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

The convex relaxation of (19) constructed by SCIP is

$$C = \mathrm{Proj}_{x,y,\theta}\left\{(x,y,z,\theta)\colon z \geq \varphi_1(x), \theta \geq \psi(y,z)\right\}.$$

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 ψ is decreasing 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 ϕ_S and $\mathrm{vex}_D[f]$ over D. From the figure we can see that, for $D = [-100,100]\times[0.01,1]$, ϕ_S and $\mathrm{vex}_D[f]$ coincide in $\{(x,y)\in D : T_{x,y}(\underline{x}) \leq \beta\underline{x}\}$, see also Figure 3. Using $D = [-100,20]\times[0.01,1]$ instead would show that ϕ_S and $\mathrm{vex}_D[f]$ coincide in all of D. The next proposition explains this behavior.

Proposition 5.1. *If $\beta\underline{x} \geq \overline{x}$, then $vex_D[f](x,y) = \phi_S(x,y)$ for all $(x,y)\in D$.*

Proof. Recall that $\mathrm{vex}_D[f](x,y) = F_{x,y}(\min\{\overline{x}, T_{x,y}(\underline{x}), \max\{x_{rl}, x_{rr}\}\})$ and $F_{x,y}(x_1) = (1-\lambda_y)\varphi_{\underline{y}}(x_1) + \lambda_y\varphi_{\overline{y}}(t_{x,y}(x_1))$. By definition, $\beta\underline{x} \leq \max\{x_{rl}, x_{rr}\}$, thus $\mathrm{vex}_D[f](x,y) = F_{x,y}(\min\{\overline{x}, T_{x,y}(\underline{x})\})$.

As $\beta\underline{x} \geq \overline{x}$, we have that $\min\{\overline{x}, T_{x,y}(\underline{x})\} \leq \beta\underline{x}$ and $t_{x,y}(\min\{\overline{x}, T_{x,y}(\underline{x})\}) \leq \beta\underline{x}$. Therefore, in $F_{x,y}(\min\{\overline{x}, T_{x,y}(\underline{x})\})$ we have that $\varphi_{\underline{y}}$ and $\varphi_{\overline{y}}$ are evaluated on points smaller than $\beta\underline{x}$, and so they are affine linear.

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

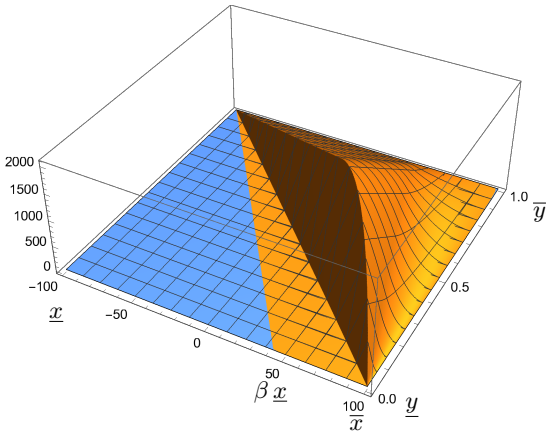

Figure 7: Difference $\mathrm{vex}_D[f](x,y) - \psi^{\mathrm{u}}(y,\varphi^{\mathrm{u}}(x))$ between the convex envelope $\mathrm{vex}_D[f](x,y)$ and the convex underestimator $\psi^{\mathrm{u}}(y,\varphi^{\mathrm{u}}(x))$ over $[-100,100]\times[0.01,1]$. The region $A^{\mathrm{u}} := \{(x,y)\in D : T_{x,y}(\underline{x}) \leq \beta\underline{x}\}$ (highlighted in blue) shows where $\mathrm{vex}_D[f](x,y)$ and $\psi^{\mathrm{u}}(y,\varphi^{\mathrm{u}}(x))$ coincide, cf. Proposition 5.1. In this region, both $\mathrm{vex}_D[f](x,y)$ and $\psi^{\mathrm{u}}(y,\varphi^{\mathrm{u}}(x))$ are linear, while the region in orange illustrates the nonlinear part of $\mathrm{vex}_D[f](x,y)$ that is tighter than $\psi^{\mathrm{u}}(y,\varphi^{\mathrm{u}}(x))$.

affine linear, we have that $(1-\lambda_y)\varphi_{\underline{y}}(T_{x,y}(\underline{x})) = \varphi_{\underline{y}}(x) - \lambda_y\varphi_{\underline{y}}(\underline{x})$. Therefore, $F_{x,y}(T_{x,y}(\underline{x})) = \varphi_{\underline{y}}(x) - \lambda_y\varphi_{\underline{y}}(\underline{x}) + \lambda_y\varphi_{\overline{y}}(\underline{x})$.

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

$$F_{x,y}(T_{x,y}(\underline{x})) = \underline{y}\varphi_1(x) + y\varphi_1(\underline{x}) - \underline{y}\varphi_1(\underline{x}).$$

Notice that $\underline{z} = \varphi_1(\underline{x})$ (as φ_1 is the convex envelope of f_1) and so

$$F_{x,y}(T_{x,y}(\underline{x})) = \underline{y}\varphi_1(x)+y\underline{z}-\underline{y}\underline{z} \leq \max\{y\underline{z} + \underline{y}\varphi_1(x) - \underline{y}\underline{z},\, y\overline{z} + \overline{y}\varphi_1(x) - \overline{y}\,\overline{z}\} = \phi_S(x,y).$$

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

The case for when $\min\{\overline{x}, T_{x,y}(\underline{x})\} = \overline{x}$ follows from a similar argument. □

We now explain the behavior from Figure 7.

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

$$\phi_S(x,y) = vex_D[f](x,y).$$

Furthermore, $vex_D[f]$ is linear in A.

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

Remark 5.3. *A similar behavior can be observed between the concave overestimator induced by SCIP and the concave envelope of $y\,sgn(x)|x|^\alpha$.* □

6 Computational Study

In this section, we present a computational study that investigates the impact of the presented convex envelope on the performance of expansion planning problem (see Section 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 $y\,\mathrm{sgn}(x)|x|^\alpha$ 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] and all heuristics[2].

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.

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

$$\mathrm{vex}_D[f](x_0, y_0) + \nabla \mathrm{vex}_D[f](x_0, y_0)^T \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \leq \pi_v - \pi_w \tag{21}$$

and

$$\mathrm{cave}_D(x_0, y_0) + \nabla \mathrm{cave}_D(x_0, y_0)^T \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \geq \pi_v - \pi_w. \tag{22}$$

Test sets. The computational study is carried out on three test sets taken from [1]: *Belgium*, *GasLib-40* and *Circuit rank*. These test sets are based upon the *Belgian* gas network from the GAMS model library[3] and the *GasLib-40* network in [22]. As we use gas instances, the exponent α is set to $\alpha = 2$ in

[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 [21].

[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`

Equation (1a). To test the model performance in diverse and severe situations both *Belgium* and *GasLib-40* instances cover a wide range of possible network demand vectors $b \in \mathbb{R}^{|\mathcal{V}|}$ and, thus, represent the whole spectrum from "easier" instances with lower demands up to "harder" instances with higher demands.

To focus also on computational difficulties that arise from more complex patterns of flow directions, the *Circuit rank* test set includes instances where the circuit rank is successively increased by adding up to ten new pipelines to the *Belgian* network, resulting in five new networks *Belgium* + (2, 4, 6, 8, 10) arcs. In total, all test sets contain 6500 instances, with *Belgium* and *GasLib-40* having each 2000 instances, and *Circuit rank* having 2500 instances.

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 Section 5).

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. To 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 $\hat{y} = \underline{y}$ or $\hat{y} = \overline{y}$, then we compute a slightly weakened gradient cut by considering the envelopes over the enlarged domain $[\underline{x}, \overline{x}] \times [\underline{y} - s, \overline{y}]$ or $[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y} + s]$, respectively. We take $s = 10^{-4}$.

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

- If the variable bound in the current LP relaxation satisfy $\overline{x} < \beta\underline{x}$, then we do not generate gradients cuts for $\mathrm{vex}_D[f](x, y)$ (Proposition 5.1).
- If the current LP solution satisfies $T_{x,y}(\underline{x}) < \beta\underline{x}$ and $\beta\underline{x} < \overline{x}$, i.e., $(x, y) \in A$, then we do not generate gradients cuts for $\mathrm{vex}_D[f](x, y)$ (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 [23] with CPLEX 12.7.1.0 as LP solver [24] and Ipopt 3.12.13 as NLP solver [25].

[4] To be released in SCIP 8.

6.2 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 that have been detected infeasible, no primal solution is known, or none of the version could increase the dual bound by more than 10^{-4}. 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 $GC\colon \mathbb{R}^3 \to [-1, 1]$ 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 [12]. 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.

	# instances	gap closed (%)
	3144	0.87
$> 0\%$ `change`	1119	2.44
$> 0\%$ `better`	1033	2.96
$> 0\%$ `worse`	86	3.82
$> 2\%$ `change`	307	6.75
$> 2\%$ `better`	271	8.74
$> 2\%$ `worse`	36	8.19
$> 10\%$ `change`	66	19.69
$> 10\%$ `better`	55	28.87
$> 10\%$ `worse`	11	16.22

Table 1: Aggregated results for the *gap closed* of the `ROOTGAP` experiment. The *gap closed* values are depicted on average for `ALL` instances, and for those instances where *Enabled* yields an improvement or deterioration of more than $0\%, 2\%$ and 10% compared to *Disabled*, called `better` or `worse`. The category `change` corresponds to the union of `better` and `worse` instances.

Aggregated results for the `ROOTGAP` experiment are shown in Table 1 and Fig-

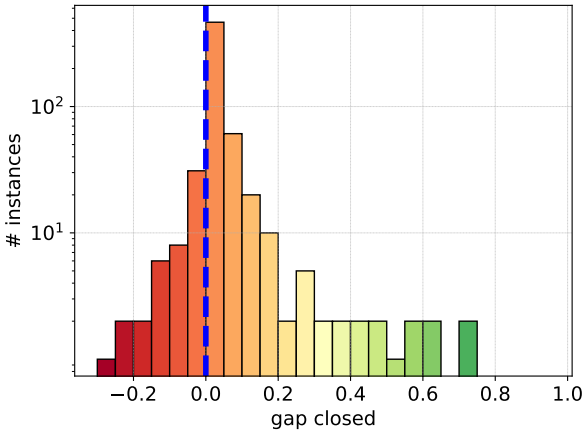

Figure 8: The *Gap closed* improvement, where each bar on the x-axis represents the relative improvement with the corresponding number of instances on the y-axis in logarithmic scale. Positive values on the x-axis represent improved dual bounds for *Enabled*, and negative values indicate better dual bounds for *Disabled*. An instance having a *gap closed* value of 0.5 (on the x-axis) implies that the gap could be closed by 50% using *Enabled* over *Disabled*. For all 3144 instances under consideration, *Enabled* yields better dual bounds on 1033 instances and *Disabled* on 86 instances.

ure 8. Table 1 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 [26]. 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.

`TREE` Experiment. Aggregated results for the `TREE` experiment are shown in Tables 2 - 4 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`, $[1, \texttt{tlim}]$, $[10, \texttt{tlim}]$, $[100, \texttt{tlim}]$, and $[1000, \texttt{tlim}]$. `ALL` consists of all instances. `ALL OPT` consists of the instances that are solved to optimality or detected to be infeasible by both settings. $[t, \texttt{tlim}]$ 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 $[t_1, \texttt{tlim}]$ are contained in $[t_2, \texttt{tlim}]$ whenever $t_1 \geq t_2$ and that one can consider the instances in $[t, \texttt{tlim}]$ 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 $[1000, \texttt{tlim}]$. This means that our separation routine renders these instances solvable but not trivially solvable.

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 2 - 4, 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 towards 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 [27] 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.[5]

[5]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.

		Enabled	*Disabled*		
	# instances	# solved	# solved	time	B&B nodes
				relative	
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	-

Table 2: Aggregated results for the TREE experiment on *Belgium*. The column "relative" reports the change of solving time and branch-and-bound nodes relative to *Enabled*.

		Enabled	*Disabled*		
	# instances	# solved	# solved	time	B&B nodes
				relative	
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	-

Table 3: Aggregated results for the TREE experiment on *GasLib-40*. The column "relative" reports the change of solving time and branch-and-bound nodes relative to *Enabled*.

		Enabled	*Disabled*		
	# instances	# solved	# solved	time	B&B nodes
				relative	
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	-

Table 4: Aggregated results for the TREE experiment on *Circuit rank*. The column "relative" reports the change of solving time and branch-and-bound nodes relative to *Enabled*.

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 [28]. 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 cru-

cial 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, [29] already report promising results for quadratic functions with absolute value terms for gas network related problems.

8 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).

References

[1] Lenz, R.: Optimization of stationary expansion planning and transient network control by mixed-integer nonlinear programming. Ph.D. thesis, Technische Universität Berlin (2021)

[2] Shiono, N., Suzuki, H.: Optimal pipe-sizing problem of tree-shaped gas distribution networks. European Journal of Operational Research **252**(2), 550–560 (2016). DOI 10.1016/j.ejor.2016.01.008

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

[4] Tawarmalani, M., Sahinidis, N.V.: Convex extensions and convex envelopes of lower semi-continuous functions. Mathematical Programming **93**, 247–263 (2002)

[5] Horst, R., Tuy, H.: Global optimization: Deterministic approaches. Springer-Verlag, Berlin Heidelberg (1990)

[6] McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part 1 – convex underestimating problems. Mathematical Programming **10**(1), 147–175 (1976)

[7] Al-Khayyal, F.A., Falk, J.E.: Jointly constrained biconvex programming. Mathematics of Operations Research **8**(2), 273–286 (1983)

[8] Sherali, H.D., Alameddine, A.: An explicit characterization of the convex envelope of a bivariate bilinear function over special polytopes. Annals of Operations Research **25**(1), 197–209 (1990)

[9] Linderoth, J.: A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. Mathematical Programming **103**(2), 251–282 (2005)

[10] Locatelli, M.: Convex envelopes of bivariate functions through the solution of kkt systems. Journal of Global Optimization pp. 1–27 (2018)

[11] Hijazi, H.: Perspective envelopes for bilinear functions. In: AIP Conference Proceedings, vol. 2070. AIP Publishing (2019)

[12] Müller, B., Muñoz, G., Gasse, M., Gleixner, A., Lodi, A., Serrano, F.: On generalized surrogate duality in mixed-integer nonlinear programming. arXiv preprint arXiv:1912.00356 (2019)

[13] Zamora, J.M., Grossmann, I.E.: A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. Journal of Global Optimization **14**(3), 217–249 (1999)

[14] Tawarmalani, M., Sahinidis, N.V.: Semidefinite relaxations of fractional programs via novel convexification techniques. Journal of Global Optimization **20**, 137–158 (2001)

[15] Jach, M., Michaels, D., Weismantel, R.: The convex envelope of (n-1)-convex functions. SIAM Journal on Optimization **19**(3), 1451–1466 (2008)

[16] 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). URL `https://nbn-resolving.de/urn:nbn:de:0297-zib-17641`

[17] Koch, T., Hiller, B., Pfetsch, M.E., Schewe, L.: Evaluating Gas Network Capacities. SIAM-MOS Series on Optimization (2015). DOI 10.1137/1.9781611973693

[18] Bragalli, C., D'Ambrosio, C., Lee, J., Lodi, A., Toth, P.: On the optimal design of water distribution networks: a practical minlp approach. Optimization and Engineering **13**(2), 219–246 (2012). DOI 10.1007/s11081-011-9141-7

[19] 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)

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

[21] Achterberg, T.: Constraint integer programming. Ph.D. thesis, Technische Universität Berlin (2007)

[22] Schmidt, M., Aßmann, D., Burlacu, R., Humpola, J., Joormann, I., Kanelakis, N., Koch, T., Oucherif, D., Pfetsch, M.E., Schewe, L., Schwarz, R., Sirvent, M.: Gaslib – a library of gas network instances. Data **2**(4) (2017). DOI 10.3390/data2040040

[23] Gleixner, A., Bastubbe, M., Eifler, L., Gally, T., Gamrath, G., Gottwald, R.L., Hendel, G., Hojny, 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). URL `https://nbn-resolving.de/urn:nbn:de:0297-zib-69361`

[24] Cplex, IBM ILOG: CPLEX: High-performance software for mathematical programming and optimization (2019). URL `https://www.ibm.com/analytics/cplex-optimizer`. Accessed in December 2019

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

[26] Gleixner, A.M., Weltge, S.: Learning and propagating lagrangian variable bounds for mixed-integer nonlinear programming. In: International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems, pp. 355–361. Springer (2013)

[27] Wilcoxon, F.: Individual comparisons by ranking methods. In: S. Kotz, N.L. Johnson (eds.) Breakthroughs in statistics, pp. 196–202. Springer (1992)

[28] Grossmann, I.E., Kravanja, Z.: Mixed-Integer Nonlinear Programming: A survey of algorithms and applications. In: L. Biegler, T. Coleman, A. Conn, F. Santosa (eds.) Large-scale Optimization with Applications, *The IMA Volumes in Mathematics and its Applications*, vol. 93, pp. 73–100. Springer, New York (1997)

[29] Liers, F., Martin, A., Merkert, M., Mertens, N., Michaels, D.: Towards the Solution of Mixed-Integer Nonlinear Optimization Problems using Simultaneous Convexification (2020). URL `https://opus4.kobv.de/opus4-trr154/frontdoor/index/index/searchtype/latest/docId/303/start/0/rows/10`