Computing Area-Tight Piece-Wise Linear Overestimators, Underestimators and Tubes for Univariate Functions

Josef Kallrath and Steffen Rebennack

Abstract We develop algorithms computing piece-wise linear, continuous overestimators, underestimators and tubes for univariate functions which minimize the area between the approximator and the function. We call such approximators area-tight. Area-tight overestimators, underestimators and tubes are of interest when solving large-scale mixed-integer non-linear programming problems. The function to be approximated is typically non-linear and non-convex; it should be univariate and discontinuous only at a finite number of points. The number of breakpoints for the approximation as well as the absolute allowable deviation of the approximator and the original function are input for the algorithms. The presented algorithms are fully automatic and make no assumptions on the shape of the function to be approximated. We provide computational tests for ten functions.

Key words: Global optimization · piece-wise linear · underestimator · overestimator · tube · are-tight · non-linear optimization · non-convexity

1 Introduction

The motivation for this publication is to follow-up on a previous work by Rebennack & Kallrath (2012, [11]) to construct over- and underestimators for one-dimensional functions. These over- and underestimators are used to replace non-linear expres-

J. Kallrath
University of Florida
Dept. of Astronomy
e-mail: kallrath@astro.ufl.edu

S. Rebennack
Colorado School of Mines
Division of Economics and Business
e-mail: srebenna@mines.edu

sions by piece-wise linear ones with the idea to approximate a non-linear (and non-convex) core and to place it into a large mixed-integer linear programming (MILP) problem. If the approximations of the feasible region and/or the objective function are constructed carefully, then the resulting MILP problem yields a lower bound (for minimization problems). In some applications, it is important to detect infeasibility of the original non-convex mixed-integer non-linear programming (MINLP) problem. Again, careful use of over- and underestimators allows for the safe conclusion of infeasibility of the original MINLP from the infeasibility of the approximate MILP problem; *cf.* [11, Section 3.3].

The concept of approximating non-linear functions by piece-wise linear ones has been around for some time. However, new developments in efficient representation of the resulting breakpoint systems (Vielma and Nemhauser, 2011, [15]) have lead to more interest in piece-wise linear approximators. Recently, Misener and Floudas (2012, [8], [9]) utilize such approximators for relaxations (underestimators) when solving mixed-integer quadratically-constrained quadratic programs.

The automatic computation of optimal breakpoint systems, however, received very little treatment in the literature. The seminal work by Rosen and Pardalos (1986, [13]) and Pardalos and Rosen (1987, [10, Chapter 8]) uses a system of equidistant breakpoints to achieve a predefined maximal deviation between a concave quadratic function and the piece-wise linear approximator. Geißler (2011,[1]) and Geißler et al. (2012, [2]) can compute piece-wise linear approximators (over- and underestimators) automatically when certain assumptions on the functions are satisfied. For more than one dimension, Misener and Floudas (2010, [7]) utilize piecewise-linear formulations via simplices; Rebennack and Kallrath (2012, [12]) use triangulations.

In Rebennack & Kallrath (2012, [11]), we minimize the number of breakpoints used to achieve a maximal deviation of δ between the piece-wise linear approximator and the original function. Furthermore, we constructed tight approximators by minimizing the maximal vertical distance between the approximator and the original function, for a given number of breakpoints. In this paper, we utilize an area-based tightness definition: allowing a maximal deviation of $\delta > 0$ and $B \in \mathbb{N} \geq 2$ breakpoints, we seek a piece-wise linear, continuous approximator which minimizes the area between the approximator and the original function. Minimizing the error between the approximator and the original function through an area-based measure is expected to produce better results (*e.g.*, tighter bounds) when replacing non-linear functions by piece-wise linear ones, compared to approaches which ignore any tightness measure.

The idea of minimizing the area between a function is briefly mentioned in Geyer et al. (2009, [3]). However, their paper does not further follow this idea but rather prefers a curvature-based approach pointing out that this is of similar quality than using vertical distances or an area-based approach. Different to our approach, they cannot guarantee the computation of an optimal breakpoint system.

The contributions of this article are as follows. For univariate functions, we develop methodologies to compute over- and underestimators as well as tubes which are (1) continuous, (2) do not deviate more than a given tolerance $\delta > 0$ from the

original function, (3) stay above (for overestimators), below (for underestimators) or a combination of both (for tubes) and are (4) area-minimizing. Thus, it is the first paper to describe a framework to automatically compute (optimal) area-minimizing breakpoint systems for univariate functions.

The remainder of the paper is organized as follows: In Section 2, we provide various definitions in the context of piece-wise linear approximators. We treat over- and underestimators in Section 3, tubes in Section 4 and approximators in Section 5. Section 6 contains our computational results. We conclude with Section 7.

2 Definitions

The original (non-linear, non-convex, continuous, and real) function to be approximated is $f(x)$ over the compactum $[X^-, X^+] \subset \mathbb{R}$. We denote by $\ell(x) : [X^-, X^+] \to \mathbb{R}$ a function approximating $f(x)$.

We start with the definition of a δ-approximator for univariate functions.

Definition 1 (δ-approximator, [11]). Let $f(x) : [X^-, X^+] \to \mathbb{R}$ be a univariate function and let scalar $\delta > 0$. A piece-wise linear, continuous function $\ell(x) : [X^-, X^+] \to \mathbb{R}$ is called a *δ-approximator* for $f(x)$, if the following property holds

$$\max_{x \in [X^-, X^+]} |\ell(x) - f(x)| \leq \delta \quad . \tag{1}$$

We require for the piece-wise linearity property of a function that the function is non-differentiable at a finite number of points. δ-over- and δ-underestimators are δ-approximators with the additional requirement to stay above or below function $f(x)$ in the domain $[X^-, X^+]$. This is formalized in

Definition 2 (δ-overestimator / δ-underestimator, [11]). We call a piece-wise linear, continuous function $\ell^+(x) : [X^-, X^+] \to \mathbb{R}$ a *δ-overestimator* for function $f(x) : [X^-, X^+] \to \mathbb{R}$, if condition (1) is satisfied along with

$$\ell^+(x) \geq f(x) \quad , \quad \forall\, x \in [X^-, X^+] \quad . \tag{2}$$

We call a piece-wise linear, continuous function $\ell^-(x)$ a *δ-underestimator* of function $f(x)$, if $-\ell^-(x)$ is a δ-overestimator of $-f(x)$.

We continue with the definition of a δ-tube.

Definition 3 (δ-tube). We call any combination of a piece-wise linear, continuous δ-overestimator $\ell^+(x)$ for function $f(x)$ and a piece-wise linear, continuous δ-underestimator $\ell^-(x)$ for $f(x)$ a *δ-tube* for $f(x)$.

The definitions of δ-approximators, δ-overestimators, δ-underestimators, and δ-tubes require piece-wise linearity and continuity. Thus, we will no longer mention these function properties explicitly in the remainder of the paper, except if we want to emphasis these two properties.

Given univariate function $f(x)$ over a compactum and the δ-tolerance, we have two desires on an automatic procedure: (1) it computes δ-approximators, δ-overestimators and/or δ-underestimators and (2) the number of required breakpoints (*i.e.*, discontinuities) is minimal. This has been achieved already (Rebennack & Kallrath, 2012, [11]). Their approach can easily be extended to compute δ-tubes which require the minimal number of breakpoints; in most cases, such optimal δ-tubes exhibit the property that the breakpoint systems of the δ-overestimator and δ-underestimator are identical, *i.e.*, both the δ-overestimator and δ-underestimator share the same discontinuities.

Vice-versa, one can provide the number of breakpoints and ask for the "tightest" possible δ-approximator, δ-overestimator, δ-underestimator, and δ-tube. In [11], the authors use an absolute function deviation error tolerance criterion as a tightness definition:

Definition 4 (absolute-error-tolerance-tightness (AETT), [11]). A δ-approximator, δ-overestimator, δ-underestimator, or δ-tube with B breakpoints for function $f(x)$ is called *tighter* (in the absolute-error-tolerance sense) than a ϑ-approximator, ϑ-overestimator , ϑ-underestimator, or ϑ-tube, respectively, with B breakpoints for function $f(x)$, if $\delta \leq \vartheta$. A δ-approximator, δ-overestimator, δ-underestimator or δ-tube with B breakpoints is called *tight* (in the absolute-error-tolerance sense) for $f(x)$, if there is no *tighter* ϑ-approximator, ϑ-overestimator, ϑ-underestimator, or ϑ-tube for $f(x)$.

In this paper, we utilize an area-based tightness definition:

Definition 5 (area-tightness (AT)). Let $\ell(x)$ be a δ-approximator, δ-overestimator, δ-underestimator, or δ-tube with B breakpoints for function $f(x)$. Further, let A_1 be the area between $\ell(x)$ and $f(x)$ over the compactum $[X^-, X^+]$. Another δ-approximator, δ-overestimator, δ-underestimator, or δ-tube with B breakpoints for function $f(x)$ and area A_2 is called *tighter* (in the area sense) than $\ell(x)$ for function $f(x)$, if $A_2 \leq A_1$. $\ell(x)$ is called *tight* (in the area sense) for $f(x)$, if there is no *tighter* δ-approximator, δ-overestimator, δ-underestimator, or δ-tube with B breakpoints for function $f(x)$.

To compute an area-tight δ-approximator, δ-overestimator, δ-underestimator, or δ-tube, we treat the error-tolerance, δ, and the number of breakpoints, B, as input parameters. Thus, we more precisely call them (δ, B)-approximator, (δ, B)-overestimator, (δ, B)-underestimator, or (δ, B)-tube.

Interestingly, AETT is preserved when shifting an absolute-error-tolerance-tight (δ, B)-approximator to obtain a (δ, B)-overestimator or (δ, B)-underestimator.

Corollary 1 ([11]). *Let $\ell(x) : [X^-, X^+] \to \mathbb{R}$ be an absolute-error-tolerance-tight (δ, B)-approximator for $f(x)$ and let $\varepsilon = 2\delta$. Then $\ell^+(x) := \ell(x) + \delta$ and $\ell^-(x) := \ell(x) - \delta$ define an absolute-error-tolerance-tight (ε, B)-underestimator and an absolute-error-tolerance-tight (ε, B)-overestimator, respectively, for $f(x)$ with the same number of breakpoints B.*

For AETT, it therefore suffice to develop one single algorithm to compute optimal (δ,B)-approximators, (δ,B)-overestimators and/or (δ,B)-underestimators; a different procedure is required for absolute-error-tolerance-tight (δ,B)-tubes. Unfortunately, AT is not preserved through (careful) shifting.

We present algorithms to compute area-tight (δ,B)-overestimators and (δ,B)-underestimators in Section 3, area-tight (δ,B)-tubes in Section 4 and area-tight (δ,B)-approximators in Section 5. However, before we proceed with the methodology, we discuss how to choose the two parameters: the absolute-error tolerance, δ, and the number of breakpoints, B. Dependent on the application, we might want to follow one of the following two paths.

If we desire to compute an approximate solution to the original MINLP problem with a specific tolerance guarantee in mind (*e.g.*, a safe gap of $\varepsilon > 0$) via piece-wise linear approximations, one needs to compute δ-approximators, δ-overestimators, δ-underestimators or δ-tubes with a certain absolute tolerance δ and apply them appropriately; *cf.* [11, Section 3.3]. In this case, we might want to proceed as follows:

1. first, compute the minimum number of breakpoints, B^*, needed to obtain a given δ-approximation (as discussed in [11]),
2. second, compute an absolute-error-tolerance-tight approximator – (ϑ,B^*)-approximator, (ϑ,B^*)-overestimator, (ϑ,B^*)-underestimator, or (ϑ,B^*)-tube – using B^* breakpoints ($\vartheta \leq \delta$; as discussed in [11]), and
3. third, compute an area-tight approximator – (ϑ,B^*)-approximator, (ϑ,B^*)-overestimator, (ϑ,B^*)-underestimator, or (ϑ,B^*)-tube.

Instead of pre-defining the tolerance (δ dependent on ε) to achieve a good lower bound for minimization problems, we might provide the number of breakpoints, B, to be spend on the piece-wise linear approximators. The number of breakpoints directly affect the model size in the MILP framework. Thus, we might want to choose the number of breakpoints in such a way that the resulting MILP problem remains efficiently solvable with (standard) solvers. Another reason for pre-defining the number of breakpoints are the use of logarithmic representations in the number of breakpoints (both in the number of binary variables and constraints involved) of the resulting breakpoint system; it is efficient to choose B as a power of 2. Given B, we would skip the first step above and compute an absolute-error-tolerance-tight approximator yielding the tolerance δ. This allows for the computation of an area-tight approximator using δ and B.

3 Computing Area-Tight (δ,B)-Overestimators and (δ,B)-Underestimators

We are given the absolute-error tolerance δ (*i.e.*, maximal vertical absolute difference between the function $f(x)$ and the approximator $\ell(x)$) and the number of breakpoints, B, for the univariate function $f(x)$ along with the closed interval $[X^-,X^+]$.

We seek to automatically compute area-tight (δ, B)-overestimators. The case of area-tight (δ, B)-underestimators follows the same logic; we discuss it in brief at the end of the section a well.

For the following discussions, we require:

- $f(x) - \delta \geq 0$ for all $x \in [X^-, X^+]$, and
- $X^- \geq 0$.

Both requirements can be achieved through a shift in either the function value direction ($f(x)$ attains a minimum in $[X^-, X^+]$, *cf.* Extreme Value Theorem) or the x-axis direction.

For our derivations, we assume that the primitive of $f(x)$ exists and we denote it by $F(x)$, for $x \in [X^-, X^+]$. We do not require its existence for our computations, though. We are interested in minimizing the area between function $f(x)$ and the piece-wise linear function $\ell^+(x)$; let $L^+(x)$ denote the primitive of $\ell^+(x)$. Therefore, we need to compute the area between the two functions. Let $x_b \in [X^-, X^+]$ denote the x-value (*i.e.*, footpoint) of the bth breakpoint and let $\ell^+(x_b)$ be its corresponding function value. Then, the area between $f(x)$ and $\ell^+(x)$ can be calculated as

$$\begin{aligned}
&\int_{X^-}^{X^+} \Big(\ell^+(x) - f(x)\Big) dx \\
&\qquad = \sum_{b=1}^{B-1} \Big[L^+(x) - F(x)\Big]_{x_b}^{x_{b+1}} \\
&\qquad = \sum_{b=1}^{B-1} \Big(L^+(x_{b+1}) - L^+(x_b)\Big) + F(x_1) - F(x_B) \\
&\qquad = \frac{1}{2} \sum_{b=1}^{B-1} \Big(\ell^+(x_{b+1}) + \ell^+(x_b)\Big)\Big(x_{b+1} - x_b\Big) + F(x_1) - F(x_B) \quad .
\end{aligned}$$

Note that the first identity is true because the approximator, $\ell^+(x)$, never crosses the function $f(x)$, *cf.* requirement (2).

We define $x_1 := X^-$ and $x_B := X^+$ implying that both $F(x_1)$ and $F(x_B)$ are fix, *i.e.*, they are constants. Thus, we are interested in minimizing the non-linear expression

$$\sum_{b=1}^{B-1} \Big(\ell^+(x_{b+1}) + \ell^+(x_b)\Big)\Big(x_{b+1} - x_b\Big) \quad .$$

Notice that we do not require the primitive (or its existence) of function $f(x)$ anymore; the numerical value of $\int_{X^-}^{X^+} f(x)dx$ suffices.

Next, we need to model the decisions on the placement of the B breakpoints, via decision variables x_b ($x_b \in [X^-, X^+]$, $x_{b+1} > x_b$, $b = 2, \ldots, B-1$), and the function values of $\ell^+(x)$ at the breakpoints, via the shit variables s_b ($s_b \in [-\delta, \delta]$, $b = 1, \ldots, B$) with respect to $f(x)$. In this context, we define

$$\phi(x_b) := f(x_b) + s_b \quad , \quad \forall\, b = 1, \ldots, B \tag{3}$$

which equals $\ell^+(x_b)$. The approximator $\ell^+(x)$ is then the corresponding interpolator between the values of $\phi(x_b)$.

Further, we need to ensure conditions (1) and (2). Both requirements lead to semi-infinite programming problems because an infinite number of (non-linear, non-convex) constraints need to hold; *cf.* Hettich and Kortanek (1993, [4]) or Lopez and Still (2007, [5]). We follow the idea of formulation OBSD as described in [11] and discretize each interval (x_{b-1}, x_b) into I equidistant grid points. Conditions (1) and (2) need then to hold on this finite grid; we increase the number of grid points dynamically until a pre-defined tolerance has been reached.

This leads us to the following (non-convex) non-linear programming (NLP) problem, computing an area-tight (δ, B)-overestimator for the continuous function $f(x)$ on the interval $[X^-, X^+]$:

$$\tilde{A}^+(\delta, B, I, M) :=$$

$$\min \sum_{b=1}^{B-1} \Big(\phi(x_{b+1}) + \phi(x_b)\Big)\Big(x_{b+1} - x_b\Big) \tag{4}$$

$$\text{s.t.} \quad x_b - x_{b-1} \geq \frac{1}{M} \quad , \quad \forall\, b = 2, \ldots, B \tag{5}$$

$$x_{bi} = x_{b-1} + \frac{i}{I+1}(x_b - x_{b-1}) \quad , \quad \forall\, b = 2, \ldots, B, \quad i = 1, \ldots, I \tag{6}$$

$$l_{bi} = \phi(x_{b-1}) + \frac{\phi(x_b) - \phi(x_{b-1})}{x_b - x_{b-1}}(x_{bi} - x_{b-1}) \quad , \quad \forall\, b = 2, \ldots, B, \quad i = 1, \ldots, I \tag{7}$$

$$l_{bi} - f(x_{bi}) \leq \delta \quad , \quad \forall\, b = 2, \ldots, B, \quad i = 1, \ldots, I \tag{8}$$

$$l_{bi} \geq f(x_{bi}) \quad , \quad \forall\, b = 2, \ldots, B, \quad i = 1, \ldots, I \tag{9}$$

$$x_1 = X^-, \quad x_B = X^+ \tag{10}$$

$$x_b \in [X^-, X^+] \quad , \quad \forall\, b = 2, \ldots, B-1 \tag{11}$$

$$x_{bi} \in [X^-, X^+] \quad , \quad \forall\, b = 2, \ldots, B, \quad i = 1, \ldots, I \tag{12}$$

$$l_{bi} \text{ free} \quad , \quad \forall\, b = 2, \ldots, B, \quad i = 1, \ldots, I \tag{13}$$

$$s_b \in [0, \delta] \quad , \quad \forall\, b = 1, \ldots, B \quad . \tag{14}$$

The logic of the constraint set (5)-(14) is as follows. Constraints (5) ensure the sorting of the breakpoints and that no two breakpoints can be identical. This becomes numerically important to avoid a division by zero when calculating the slope of the approximator $\ell^+(x)$. The value of the constant M needs to be chosen carefully in order to avoid exclusion of an optimal distribution of the breakpoints. Actually, is it non-trivial to mathematical (and computational) safely conclude what a sufficiently large value for M is. Constraints (6) model the I grid points, x_{bi}, for the interval (x_{b-1}, x_b). These grid points are the discretization introduced in order to ensure that (I) the maximal vertical distance between function $f(x)$ and the approximator $\ell^+(x)$ is at most δ, as required in (1) and modeled via (7) & (8), and that (II) approxima-

tor $\ell^+(x)$ stays above function $f(x)$ as required in (2) and modeled via (7) & (9). Constraints (10)-(14) model the variables' domain.

The mathematical model (4)-(14) is non-linear, non-convex and continuous: It consists of $2B+2(B-1)I-2$ continuous variables and $B+4(B-1)I-1$ constraints; the objective function (4) as well as constraints (7)-(9) are non-convex.

If the NLP (4)-(14) is infeasible, then there are two possibilities: either M is too small or the combination of δ and B does not allow for the existence of a (δ,B)-overestimator.

The idea of the objective function (4) is intuitive: We minimize the area of the approximator $\ell^+(x)$ and the x-axis; constraints (9) ensure that $\ell^+(x)$ always stays above function $f(x)$. Given a sufficiently large value for M denoted by M^*, we can recover a lower bound $\underline{A}^+$ on the area A between the approximator $\ell^+(x)$ and the original function $f(x)$ via

$$\underline{A}^+ = \tfrac{1}{2}\tilde{A}^+(\delta,B,I,M^*)+F(x_1)-F(x_B) \quad . \tag{15}$$

Equation (15) constitutes a lower bound on the area A because both conditions (1) and (2) are relaxed; they hold only on a finite number of (grid) points.

After solving (4)-(14) to (local or global) optimality, we solve (to global optimality)

$$\mu^+(I) := \max_{b=2,\ldots,B} \mu_b^+(I) := \max_{b=2,\ldots,B} \max_{x\in[x_{b-1},x_b]} \Big(\ell^+(x)-f(x)\Big) \tag{16}$$

in order to compute the maximal vertical deviation between $f(x)$ and the computed approximator $\ell^+(x)$ in the interval $[X^-,X^+]$. If

$$\mu^+(I) \leq \delta \quad , \tag{17}$$

then condition (1) holds true and the computed $\ell^+(x)$ defines a (δ,B)-approximator for $f(x)$.

We further need to check if $\ell^+(x)$ is below function $f(x)$ somewhere in the interval (X^-,X^+). Therefore, we solve (to global optimality)

$$\psi^+(I) := \min_{b=2,\ldots,B} \psi_b^+(I) := \min_{b=2,\ldots,B} \min_{x\in[x_{b-1},x_b]} \Big(\ell^+(x)-f(x)\Big) \quad . \tag{18}$$

If

$$\psi(I) \geq 0 \quad , \tag{19}$$

then condition (2) holds true. If both (17) and (19) are satisfied, then $\ell^+(x)$ defines an area-tight (δ,B)-overestimator for $f(x)$ with $A=\underline{A}^+$.

If (17) or (19) are violated by more than a pre-defined tolerance $\eta>0$, then we increase the number of grid points, I, and re-solve (4)-(14) as well as (16) and (18). For any desired precision $\eta>0$, this process, of increasing I, is finite (granted that the NLP problems can be solved to global optimality).

Corollary 2. *Let $f(x)$ be a continuous function on $[X^-,X^+]$, $\delta > 0$ and $B \in \mathbb{N} \geq 2$ be fixed. Then, for each $\eta > 0$, there exists a finite I^*, such that $\mu(I^*) \leq \delta + \eta$ and $\psi(I^*) \geq -\eta$, given that there exists a (δ,B)-overestimator for $f(x)$.*

The proof of Corollary 2 is based on the continuity of $f(x)$ over a compactum and follows from Rebennack & Kallrath [11, Corollary 6].

Following the same logic as for the area-tight (δ,B)-overestimator, we compute an area-tight (δ,B)-underestimator, $\ell^-(x)$, for $f(x)$ on the interval $[X^-,X^+]$:

$$\tilde{A}^-(\delta,B,I,M) :=$$

$$\max \sum_{b=1}^{B-1} \Big(\phi(x_{b+1}) + \phi(x_b)\Big)\Big(x_{b+1} - x_b\Big) \tag{20}$$

$$\text{s.t.} \quad (5)-(7),(10)-(13) \tag{21}$$

$$f(x_{bi}) - l_{bi} \leq \delta \quad , \quad \forall\, b = 2,\ldots,B, \quad i = 1,\ldots,I \tag{22}$$

$$l_{bi} \leq f(x_{bi}) \quad , \quad \forall\, b = 2,\ldots,B, \quad i = 1,\ldots,I \tag{23}$$

$$s_b \in [-\delta, 0] \quad , \quad \forall\, b = 1,\ldots,B \quad . \tag{24}$$

Analogously, the condition (1) reads for underestimators

$$\mu^-(I) := \max_{b=2,\ldots,B} \mu_b^-(I) := \max_{b=2,\ldots,B} \max_{x\in[x_{b-1},x_b]} \Big(f(x) - \ell^-(x)\Big) \tag{25}$$

and (2) is

$$\psi^-(I) := \min_{b=2,\ldots,B} \psi_b^-(I) := \min_{b=2,\ldots,B} \min_{x\in[x_{b-1},x_b]} \Big(f(x) - \ell^-(x)\Big) \quad . \tag{26}$$

Function $\ell^-(x)$ defines an area-tight (δ,B)-underestimator for $f(x)$ with area

$$A = \tfrac{1}{2}\tilde{A}^-(\delta,B,I,M) + F(x_1) - F(x_B) \quad ,$$

if both

$$\mu^-(I) \leq \delta \qquad \text{and} \qquad \psi^-(I) \geq 0 \quad . \tag{27}$$

Corollary 2 reads now

Corollary 3. *Let $f(x)$ be a continuous function on $[X^-,X^+]$, $\delta > 0$ and $B \in \mathbb{N} \geq 2$ be fixed. Then, for each $\eta > 0$, there exists a finite I^*, such that $\mu^-(I^*) \leq \delta + \eta$ and $\psi^-(I^*) \geq -\eta$, given that there exists a (δ,B)-underestimator for $f(x)$.*

4 Computing an Area-Tight (δ,B)-Tube: (δ,B)-Overestimators and (δ,B)-Underestimators Sharing the Same Breakpoint System

Recall that the purpose of piece-wise linear approximations of functions is to replace a non-liner system of constraints or objective function by MILP constructs to be placed in a MILP framework. Therefore, consider a non-convex, continuous, univariate function $f(x)$ which appears as an equation

$$f(x) = b \quad , \quad x \in [X^-, X^+]$$

in the constraints of the MINLP problem to be approximated. In this case, one would compute an area-tight (δ,B)-overestimator, $\ell^+(x)$, and an area-tight (δ,B)-underestimator, $\ell^-(x)$, for $f(x)$. When doing so, there is no guarantee that the breakpoint systems of $\ell^+(x)$ and $\ell^-(x)$ are identical. Most likely, we would require $2(B-1)$ breakpoints for the resulting δ-tube. Notice that the resulting tube might not be an area-tight $(\delta,2B-2)$-tube. For a given number of breakpoints, B, an are-tight (δ,B)-tube can be calculated when the (δ,B)-overestimator and the (δ,B)-underestimator share the same breakpoint system. Notice that the resulting (δ,B)-overestimator and (δ,B)-underestimator might not be area-tight, even though the (δ,B)-tube is.

Just like in the previous section, for notational convenience, we assume that

- $f(x)-\delta \geq 0$ for all $x \in [X^-,X^+]$, and
- $X^- \geq 0$.

For (δ,B)-overestimator, $\ell^+(x)$, and (δ,B)-underestimator, $\ell^-(x)$, sharing the same B breakpoints at x_b, the area of the resulting (δ,B)-tube is derived through

$$\begin{aligned}
&\int_{X^-}^{X^+} \Big(\ell^+(x)-\ell^-(x)\Big)dx \\
&\qquad = \sum_{b=1}^{B-1} \Big[L^+(x)-L^-(x)\Big]_{x_b}^{x_{b+1}} \\
&\qquad = \sum_{b=1}^{B-1} \Big(L^+(x_{b+1})-L^+(x_b)-L^-(x_{b+1})+L^-(x_b)\Big) \\
&\qquad = \frac{1}{2}\sum_{b=1}^{B-1} \Big(\ell^+(x_{b+1})+\ell^+(x_b)\Big)\Big(x_{b+1}-x_b\Big) \\
&\qquad\qquad\qquad -\frac{1}{2}\sum_{b=1}^{B-1} \Big(\ell^-(x_{b+1})+\ell^-(x_b)\Big)\Big(x_{b+1}-x_b\Big) \quad .
\end{aligned}$$

Similar to (3), we define

$$\phi^+(x_b) := f(x_b)+s_b^+ \quad \text{and} \quad \phi^-(x_b) := f(x_b)+s_b^- \quad , \quad \forall\, b=1,\ldots,B \quad .$$

Following the idea of formulation (4)-(14), we obtain the following continuous, non-convex NLP problem, computing an area-tight (δ,B)-tube for the continuous function $f(x)$ on the interval $[X^-,X^+]$:

$$\tilde{A}^{\pm}(\delta,B,I,M) :=$$

$$\min \ \frac{1}{2}\sum_{b=1}^{B-1}\Big(\phi^+(x_{b+1})+\phi^+(x_b)-\phi^-(x_{b+1})-\phi^-(x_b)\Big)\Big(x_{b+1}-x_b\Big) \tag{28}$$

$$\text{s.t.} \quad (5),(6),(10)-(12) \tag{29}$$

$$l_{bi}^+ = \phi^+(x_{b-1})+\frac{\phi^+(x_b)-\phi^+(x_{b-1})}{x_b-x_{b-1}}(x_{bi}-x_{b-1}) \quad , \quad \forall\, b=2,\ldots,B, \quad i=1,\ldots,I \tag{30}$$

$$l_{bi}^+ - f(x_{bi}) \le \delta \quad , \quad \forall\, b=2,\ldots,B, \quad i=1,\ldots,I \tag{31}$$

$$l_{bi}^+ \ge f(x_{bi}) \quad , \quad \forall\, b=2,\ldots,B, \quad i=1,\ldots,I \tag{32}$$

$$l_{bi}^- = \phi^-(x_{b-1})+\frac{\phi^-(x_b)-\phi^-(x_{b-1})}{x_b-x_{b-1}}(x_{bi}-x_{b-1}) \quad , \quad \forall\, b=2,\ldots,B, \quad i=1,\ldots,I \tag{33}$$

$$f(x_{bi}) - l_{bi}^- \le \delta \quad , \quad \forall\, b=2,\ldots,B, \quad i=1,\ldots,I \tag{34}$$

$$l_{bi}^- \le f(x_{bi}) \quad , \quad \forall\, b=2,\ldots,B, \quad i=1,\ldots,I \tag{35}$$

$$l_{bi}^+,\ l_{bi}^- \text{ free} \quad , \quad \forall\, b=2,\ldots,B, \quad i=1,\ldots,I \tag{36}$$

$$s_b^+ \in [0,\delta] \quad , \quad s_b^- \in [-\delta,0], \quad \forall\, b=1,\ldots,B \quad . \tag{37}$$

Constraint group (29) models the breakpoint system, constraints (30)-(32) model the overestimator and (33)-(35) the underestimator.

The computed $\ell_+(x)$ defines a (δ,B)-overestimator, if both (17) and (19) hold true; $\ell_-(x)$ is a (δ,B)-underestimator, if both conditions in (27) hold. If all four conditions are satisfied, then $\ell^+(x)$ and $\ell^-(x)$ define an area-tight (δ,B)-tube for $f(x)$ on $[X^-,X^+]$ with area $\tilde{A}^{\pm}(\delta,B,I,M)$; otherwise, if at least one of the four conditions is violated, then the grid size I needs to be increased.

We also have a finite convergence argument for tubes.

Corollary 4. *Let $f(x)$ be a continuous function on $[X^-,X^+]$, $\delta>0$ and $B\in\mathbb{N}\ge 2$ be fixed. Then, for each $\eta>0$, there exists a finite I^*, such that $\max\{\mu^+(I^*),\mu^-(I^*)\}\le\delta+\eta$ and $\min\{\psi^+(I^*),\psi^-(I^*)\}\ge-\eta$, given that there exists a (δ,B)-tube for $f(x)$.*

5 Computing Area-Tight (δ,B)-Approximators

δ-approximators play the central role in the methodology developed by Rebennack & Kallrath (2012, [11]), because they allow for the efficient computation of (absolute-error-tolerance-tightness) δ-overestimators and δ-underestimators via a

simple function value shift; minimality in the number of breakpoints required is preserved as well. The case for area-tight (δ, B)-approximators is different: AT is not preserved after a shifting operation.

Over-, underestimators and tubes are important constructs when replacing NLP problems; approximators are not equally important, as they do not allow for the computation of safe bounds and do not allow for infeasiblity detection. Thus, we leave it at a sketch of the idea on how to compute an area-tight (δ, B)-approximator.

Approximators can intersect with the function $f(x)$, unlike over- and underestimators. This poses a challenge, when calculating the area between the approximator and the function. We use the following idea: given that we are working with a grid (the I discrete points) on the x-axis, we evaluate the relative position of the approximator $\ell(x)$ to the function $f(x)$ at these grid points by introducing the binary decision variables γ_{bi} with

$$-\delta\big(1-\gamma_{bi}\big) \leq f(x_{bi}) - l_{bi} \leq \delta\gamma_{bi} \quad , \quad \forall\, b = 2,\ldots,B, \quad i = 1,\ldots,I \quad .$$

If $f(x)$ is above (below) the approximator $\ell(x)$ at point x_{bi}, *i.e.*, $f(x_{bi}) > l_{bi}$ ($f(x_{bi}) < l_{bi}$), then $\gamma_{bi} = 1$ ($\gamma_{bi} = 0$).

We consider only the case in which the primitive of function of f exists. We distinguish three cases on the relative position of the approximator to the function $f(x)$, to calculate an approximation of the area between $f(x)$ and $\ell(x)$

I: $\gamma_{bi} = \gamma_{b,i+1} = 1$

$$F(x_{b,i+1}) - F(x_{bi}) - L(x_{b,i+1}) + L(x_{bi})$$

this formula is precise if $f(x) \geq \ell(x)$ for all $x \in [x_{bi}, x_{b,i+1}]$

II: $\gamma_{bi} = \gamma_{b,i+1} = 0$

$$-F(x_{b,i+1}) + F(x_{bi}) + L(x_{b,i+1}) - L(x_{bi})$$

this formula is precise if $f(x) \leq \ell(x)$ for all $x \in [x_{bi}, x_{b,i+1}]$

III: $\gamma_{bi} \neq \gamma_{b,i+1}$ the approximator ℓ intersects with the function f at least once in the interval $x \in [x_{bi}, x_{b,i+1}]$, we assign the area a value of 0.

The three cases above are restricted to the intervals $[x_{b1}, x_{bI}]$, $b = 2,\ldots,B$, and do neither consider the interval $[x_{b-1}, x_{b1}]$ nor $[x_{bI}, x_b]$ located around the breakpoints, $b = 1,\ldots,B$. Therefore, we introduce the binary decision variable γ_b with

$$-\delta\big(1-\gamma_b\big) \leq s_b \leq \delta\gamma_b \quad , \quad \forall\, b = 1,\ldots,B, \quad .$$

and derive the area of the intervals using the three cases above analogously.

The resulting mathematical programming problem is a mixed-integer non-linear programming problem, which is non-convex. The number of binary variables depends on the number of breakpoints, B, and the grid size, I. Therefore, we expect that the computation of area-tight (δ, B)-approximators is computationally much harder than the computation of area-tight (δ, B)-overestimators or area-tight (δ, B)-underestimators.

After the resulting MINLP has been solved, we check if the continuums-condition (1) is satisfied, via the solution of the global optimization problem

$$\mu^{\pm}(I) := \max_{b=2,\ldots,B} \mu_b(I) := \max_{b=2,\ldots,B} \max_{x\in[x_{b-1},x_b]} \Big|\ell(x) - f(x)\Big| \quad .$$

If $\mu(I) > \delta$, then we increase I and start-over; otherwise, $\ell(x)$ is a (δ,B)-approximator. The area computed as described above defines a lower bound on the area of an area-tight (δ,B)-approximator; an upper bound is obtained by evaluating the area between the calculated $\ell(x)$ and $f(x)$. If the lower and the upper bound on the area are close enough together, then we stop, otherwise we increase I further.

6 Computational Results

We execute our computational tests on an Intel(R) i7 @ 2.40Ghz with 8 GB RAM running 64-bit Windows 7. We use GAMS version 23.8 and solve all non-convex NLP problems with the global solver LindoGLOBAL [14] to an absolute gap (*i.e.*, upper bound minus lower bound) of 10^{-5}.

For our computational tests, we made the following selection for the parameters I, M and η. We start with a grid size of $I = 2$ and update the number of grid points according to the following formula

$$\max\{\lfloor 1.5I \rfloor, I+1\} \quad .$$

We choose $M = 10^{-5}$ as well as $\eta = 0.001$. We use the 10 univariate functions, taken from the literature, as summarized in Table 1.

Table 2 summarized the computational results for area-tight (δ,B)-overestimators. We make the following observations: (I) area-tight (δ,B)-overestimators can only be computed for a few number of breakpoints; (II) the number of discretization points (*i.e.*, I) required to ensure a maximal violation of 0.001 of condition (2) (II.1) varies widely among the tested functions: if the function is convex (*e.g.*, function 01), then any discretization suffices, and (II.2) decreases with an increase in the number of breakpoints; (III) the computational time tends to increase exponentially in the number of breakpoints.

The computational results for area-tight (δ,B)-underestimators are provided in Table 3. The concavity of function 02 makes it possible to compute area-tight (δ,B)-underestimators for up to 15 breakpoints within the time limit. Functions 08 and 09 are difficult to tightly underestimate: the value of M needs to be chosen carefully; local solvers might easily miss a global optimum for (20)-(24).

Results for area-tight (δ,B)-tubes for the ten test functions are given in Table 4. The column labeled "$\underline{A}^{+} + \underline{A}^{-}$" reports on the sum of the area of the corresponding area-tight (δ,B)-overestimator and area-tight (δ,B)-underestimator, which is a lower bound on the area of a (δ,B)-tube. Further, $\mu^{\pm} := \max\{\mu^{+},\mu^{-}\}$ provides the maximal absolute vertical deviation of the tube to the original function $f(x)$. In-

Table 1: One-dimensional test functions taken from Rebennack and Kallrath (2013, [11]).

#	$\mathbf{f}(\mathbf{x})$	$\mathbf{X}_-$	$\mathbf{X}_+$	**Comment**
01	x^2	-3.5	3.5	convex function; axial symmetric at $x=0$
02	$\ln x$	1	32	concave function
03	$\sin x$	0	2π	point symmetric at $x=\pi$
04	$\tanh(x)$	-5	5	strictly monotonically increasing; point symmetric at $x=0$
05	$\frac{\sin(x)}{x}$	1	12	for numerical stability reason we avoid the removable singularity and the oscillation at 0, the two local minima have an absolute function value difference of ≈ 0.126
06	$2x^2+x^3$	-2.5	2.5	in $(-\infty,\infty)$, there is one local minimum at $x=0$ and one local maximum at $x=\frac{4}{3}$
07	$e^{-x}\sin(x)$	-4	4	one global minimum ($x_m \approx -2.356$ with $f(x_m)\approx -7.460$)
08	$e^{-100(x-2)^2}$	0	3	a normal distribution with a sharp peak at x=2
09	$1.03e^{-100(x-1.2)^2}+e^{-100(x-2)^2}$	0	3	the sum of two Gaussians, with two slightly different maxima (their absolute function value difference is ≈ 0.030)
10	Maranas & Floudas (1994, [6])	0	2π	three local minima (the absolute function value difference of the two smallest local minima is ≈ 0.031)

terestingly, the area of an area-tight (δ,B)-tube is only marginally larger (if at all), for the tested functions, compared to the area provided by combining an area-tight (δ,B)-overestimator with an area-tight (δ,B)-underestimator, while the number of breakpoints for the area-tight (δ,B)-tubes is almost half compared to the combination of an area-tight (δ,B)-overestimator with an area-tight (δ,B)-underestimator. Computing area-tight (δ,B)-tubes is computationally more challenging than computing area-tight (δ,B)-overestimators and area-tight (δ,B)-underestimators. However, it remains computational tractable to compute area-tight (δ,B)-tubes for a small number of breakpoints.

Figure 1 shows plots of the ten test functions together with an area-tight (δ,B)-overestimator, (δ,B)-underestimator or (δ,B)-tube. The presented over-, underestimators and tubes correspond to the results of Tables 2, 3 and 4.

7 Conclusions

In this paper, we extend the literature on methodologies which automatically compute optimal piece-wise linear overestimators, underestimators and tubes for uni-

Table 2: Area-tight (δ, B)-overestimators for the functions provided in Table 1.

#	**B**	δ	$\underline{A}^+$	ψ^+	μ^+	I	**sec.**
01	3	3.10	14.2917	0.0000000	3.063	2	0.19
	4	1.50	6.3519	0.0000000	1.361	2	0.91
	5	1.10	3.5729	0.0000000	0.766	2	24.35
	6	1.10	2.2867	0.0000000	0.490	2	329.53
	7	0.40	1.5880	0.0000000	0.340	2	0.78
	8	0.40	–	–	–	2	3600.07†
02	3	1.00	2.4186	-0.0005192	0.900	9	4.38
	4	0.85	1.1780	-0.0005961	0.494	9	154.21
	5	0.45	–	–	–	2	3600.10†
03	3	1.50	3.4820	-0.0005656	1.365	28	12.37
	4	0.40	0.7448	-0.0002769	0.278	28	50.67
	5	0.40	0.4484	-0.0004956	0.311	28	1348.89
	6	0.40	0.2958	-0.0006979	0.125	13	5965.65
	7	0.40	–	–	–	3	7081.38†
04	3	1.00	3.2294	-0.0002624	0.958	13	4.07
	4	0.30	0.4874	-0.0007642	0.192	3	3.42
	5	0.20	0.2660	-0.0002292	0.172	13	136.22
	6	0.20	0.1819†	-0.0010273†	–	19	7808.39†
05	3	1.00	1.4856	-0.0007117	0.301	42	30.76
	4	0.40	0.5659	-0.0004862	0.106	13	28.47
	5	0.40	0.3583	-0.0002181	0.102	13	412.04
	6	0.40	0.1849	-0.0007009	0.049	9	1650.26
	7	0.40	0.1395†	-0.0041407†	–	6	6894.49†
06	3	5.00	8.4034	-0.0004046	3.959	28	11.81
	4	4.50	4.5613	0.0000000	4.369	63	1035.67
	5	4.50	3.1492†	-0.0027268†	–	42	9040.39†
07	3	30.00	17.0289	-0.0005812	7.490	94	87.63
	4	10.00	11.9770	-0.0002707	9.569	42	846.74
	5	4.00	4.8733	-0.0003621	3.603	28	5184.76
	6	4.00	2.7909†	-0.0053520†	–	3	5223.90†
08	3	1.00	1.3130‡	-0.0110870‡	–	141	562.27‡
	4	1.00	0.4476	0.0000000	0.785	63	6338.37
	5	1.00	0.0626	-0.0006100	0.237	13	622.74
	6	1.00	0.0376†	-0.0016832†	–	28	6845.49†
09	3	1.00	1.9998‡	-0.0077818‡	–	141	1262.31‡
	4	1.00	1.3293†	-0.0862732†	–	42	7569.87†
10	3	4.00	10.2380‡	-0.0069331‡	–	141	4376.43‡
	4	4.00	8.6188†	-0.1971054†	–	19	9491.13†

†: out of time (time limit per model is 3600 seconds)

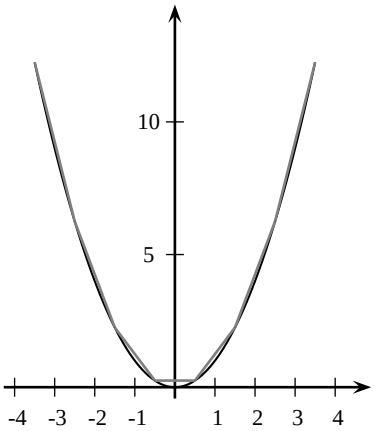
‡: model size exceeds license limits

Table 3: Area-tight (δ, B)-underestimators for the functions provided in Table 1.

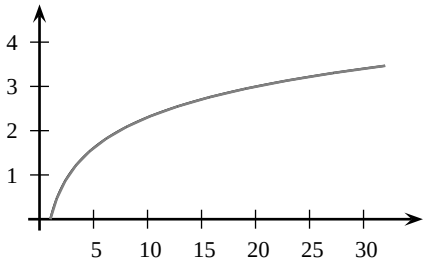
#	**B**	δ	$\underline{A}^-$	ψ^-	μ^-	I	**sec.**
01	3	3.10	7.1458	-0.0001151	3.100	3	0.73
	4	1.50	3.1759	-0.0000429	1.376	9	41.06
	5	1.10	1.7801†	-0.0009104†	–	28	4542.38†
02	3	1.00	5.9903	0.0000000	0.564	2	0.33
	4	0.85	2.6130	0.0000000	0.319	2	2.03
	5	0.45	1.4598	0.0000000	0.205	2	28.95
	6	0.45	0.9312	0.0000000	0.143	2	5.80
	7	0.25	0.6455	0.0000000	0.106	2	51.80
	8	0.25	0.4738	0.0000000	0.081	2	61.68
	9	0.25	0.3625	0.0000000	0.064	2	44.42
	10	0.25	0.2863	0.0000000	0.052	2	67.59
	11	0.25	0.2318	0.0000000	0.043	2	8.77
	12	0.25	0.1915	0.0000000	0.036	2	299.95
	13	0.25	0.1609	0.0000000	0.031	2	380.18
	14	0.25	0.1371	0.0000000	0.027	2	858.85
	15	0.25	0.1182	0.0000000	0.023	2	526.15
	16	0.25	–	–	–	2	3601.02†
03	3	1.50	3.4820	-0.0005656	1.365	28	13.46
	4	0.40	0.7448	-0.0002769	0.278	28	62.06
	5	0.40	0.4484	-0.0004956	0.311	28	1118.99
	6	0.40	0.2958†	-0.0027497†	–	13	7059.87†
04	3	1.00	3.2294	-0.0002941	0.958	13	3.03
	4	0.30	0.4874	-0.0007642	0.192	3	3.06
	5	0.20	0.2661	-0.0000696	0.180	19	202.68
	6	0.20	0.1819†	-0.0010272†	–	19	6774.00†
05	3	1.00	1.0176	-0.0006447	0.285	9	2.71
	4	0.40	0.3514	-0.0008220	0.157	13	56.77
	5	0.40	0.2615	-0.0002037	0.150	19	1854.52
	6	0.40	–	–	–	3	6561.11†
06	3	5.00	7.1298	-0.0005952	3.779	3	1.39
	4	4.50	4.0965	-0.0007573	4.351	63	1319.01
	5	4.50	2.0713†	-0.0048858†	–	28	7504.65†
07	3	30.00	20.1332‡	-0.0085689‡	–	141	196.96‡
	4	10.00	6.3694†	-0.0016387†	–	94	6880.37†
08	3	1.00	0.1772	0.0000000	1.000	3	0.86
	4	1.00	0.1764	-0.0009344	0.997	4	67.11
	5	1.00	0.0205	0.0000000	0.108	6	425.71
	6	1.00	0.0142	-0.0000999	0.106	4	354.25
	7	1.00	0.0142	-0.0000999	0.106	4	731.74
	8	1.00	0.0109†	-0.0066468†	–	9	6050.35†
09	3	1.00	0.3598	0.0000000	1.030	6	4.81
	4	1.00	0.3597	-0.0001966	1.030	9	646.77
	5	1.00	0.1984	0.0000000	1.000	6	1832.71
	6	1.00	0.1966	-0.0004934	1.030	4	2752.09
	7	1.00	–	–	–	2	3601.98†
10	3	4.00	7.9921‡	-0.0017494‡	–	141	5844.36‡
	4	4.00	6.218†	-0.2982953†	–	13	5864.30†

†: out of time (time limit per model is 3600 seconds)

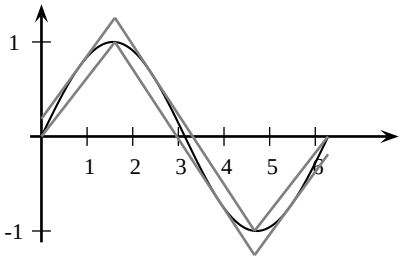
‡: model size exceeds license limits



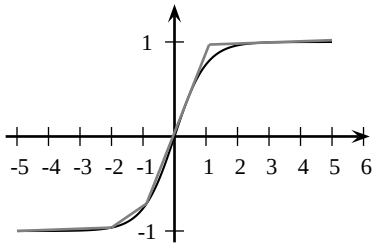
(a) 01: area-tight $(0.4, 8)$-overestimator



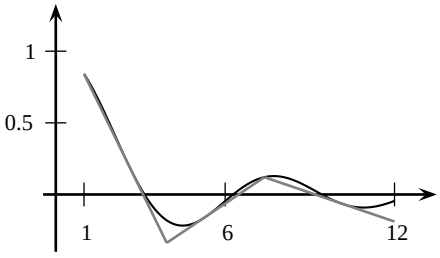
(b) 02: area-tight $(0.25, 15)$-underestimator



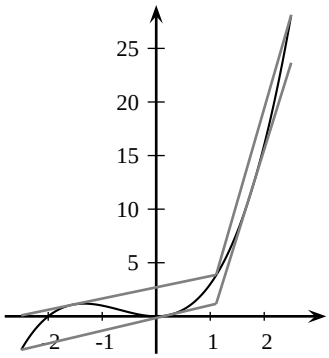
(c) 03: area-tight $(0.4, 4)$-tube



(d) 04: area-tight $(0.2, 5)$-overestimator



(e) 05: area-tight $(0.4, 4)$-underestimator



(f) 06: area-tight $(5, 3)$-tube

Fig. 1: Continued.

Table 4: Area-tight (δ, B)-tubes for the functions provided in Table 1.

#	**B**	δ	$\underline{A}^+ + \underline{A}^-$	$\underline{A}^\pm$	ψ^+	ψ^-	$\mu^\pm$	I	**sec.**
01	3	3.10	21.4375	21.4375	0.0000000	-0.0001148	3.100	3	1.40
	4	1.50	9.5278	9.5278	0.0000000	-0.0000105	1.369	9	178.39
	5	1.10	♯	5.3594†	0.0000000†	-0.0156250†	–	9	3707.36†
02	3	1.00	8.4089	8.4292	-0.0008616	0.0000000	0.788	13	19.56
	4	0.85	3.7910	3.7946	-0.0004650	0.0000000	0.449	9	257.74
	5	0.45	♯	2.1479	-0.0003956	0.0000000	0.282	9	437.53
	6	0.45	♯	1.3279†	-0.0016468†	0.0000000†	–	4	3833.89†
03	3	1.50	6.9639	7.3622	-0.0000396	-0.0004090	1.500	42	91.53
	4	0.40	1.4896	1.5018	-0.0006088	-0.0006088	0.257	19	141.01
	5	0.40	0.8967	1.0616†	-0.0020084†	-0.0088723†	–	9	4516.01†
04	3	1.00	6.4588	7.9908	-0.0006473	-0.0002773	1.000	42	85.96
	4	0.30	0.9748	0.9967	-0.0006409	-0.0006409	0.154	6	42.62
	5	0.20	0.5321	0.7070	-0.0002143	-0.0006732	0.174	13	1858.18
	6	0.20	♯	–	–	–	–	2	3600.11†
05	3	1.00	2.5032	2.6914	-0.0006133	-0.0003608	0.453	42	70.57
	4	0.40	0.9173	0.9235	-0.0006387	0.0000000	0.157	13	115.89
	5	0.40	0.6198	0.6192†	-0.0028052†	-0.0007355†	–	13	5500.06†
06	3	5.00	15.5331	15.6470	-0.0007989	-0.0008506	4.466	63	131.09
	4	4.50	8.6578	10.2935†	0.0000000†	-0.0039028†	–	63	5896.67†
07	3	30.00	♯	37.492‡	-0.0006538‡	-0.0073744‡	–	141	494.11‡
	4	10.00	♯	19.2815	-0.0003845	-0.0009017	10.000	42	3810.09
	5	4.00	♯	8.518†	-0.0241269†	-0.0248196†	–	13	9022.88†
08	3	1.00	♯	1.4903‡	-0.0110513‡	0.0000000‡	–	141	3660.14‡
	4	1.00	0.6249	0.6221†	-0.1077110†	0.0000000†	–	42	3889.13†
09	3	1.00	♯	2.3596†	-0.0152941†	0.0000000†	–	94	5729.48†
	4	1.00	♯	1.7519†	-0.0862732†	0.0000000†	–	42	7235.23†
10	3	4.00	♯	18.6457†	-0.0125147†	-0.0071669†	–	94	8319.97†
	4	4.00	♯	13.1937†	-1.0792070†	-0.0033989†	–	9	4397.01†

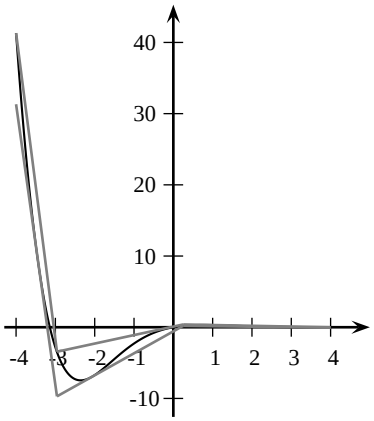
†: out of time (time limit per model is 3600 seconds)

‡: model size exceeds license limits

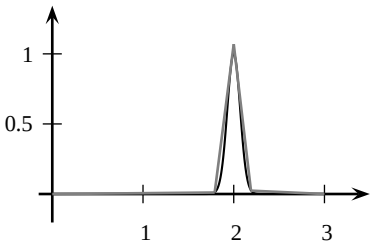
♯: over- and/or underestimator problem was not solved to global optimality

variate functions. The computed approximators are optimal among all piece-wise linear, continuous functions in the sense that they minimize the area between the function and the approximator. Our methodology for computing area-tight (δ, B)-overestimators, (δ, B)-underestimators and (δ, B)-tubes require the solution of a series of continuous, non-linear, and non-convex mathematical programming problems.

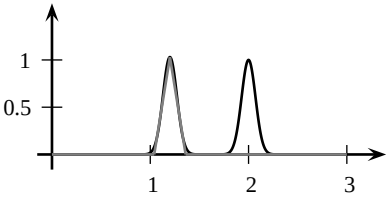
The computational tests reveal that it is worth-while to compute area-tight (δ, B)-tubes which share the same breakpoint system, rather than computing (δ, B)-overestimators and (δ, B)-underestimators individually, if tubes are desired.



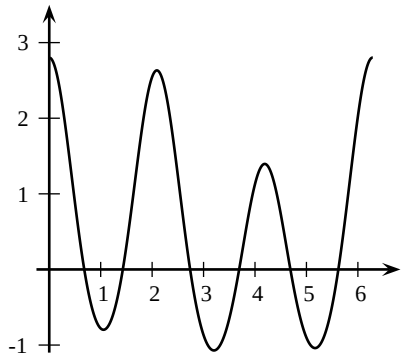
(g) 07: area-tight $(10,4)$-tube



(h) 08: area-tight $(1,5)$-overestimator



(i) 09: area-tight $(1,5)$-underestimator



(j) 10: no area-tight approximation

Fig. 1: The ten univariate functions together with computed (δ,B)-overestimator, (δ,B)-underestimator, or (δ,B)-tube.
— original function $f(x)$
— approximator function $\ell^+(x)$, $\ell^-(x)$, or $\ell^\pm(x)$

References

1. B. Geißler. *Towards Globally Optimal Solutions for MINLPs by Discretization Techniques with Applications in Gas Network Optimization*. Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen-Nürnberg, Germany, 2011.
2. B. Geißler, A. Martin, A. Morsi, and L. Schewe. Using Piecewise Linear Functions for Solving MINLPs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 287–314. Springer, 2012.

3. A. Geyer, M. Hanke, and A. Weissensteiner. Life-cycle Asset Allocation and Consumption Separable using Stochastic Linear Programming. *The Journal of Computational Finance*, 12:29–50, 2009.
4. R. Hettich and K. O. Kortanek. Semi-infinite Programming. *SIAM Review*, 35:380–429, 1993.
5. M. Lopez and G. Still. Semi-infinite Programming. *European Journal of Operational Research*, 180:491–518, 2007.
6. C. Maranas and C. A. Floudas. Global Minimum Potential Energy Conformations of Small Molecules. *Journal of Global Optimization*, 4:135–170, 1994.
7. R. Misener and C. A. Floudas. Piecewise-Linear Approximations of Multidimensional Functions. *Journal of Optimization Theory and Applications*, 145:120–147, 2010.
8. R. Misener and C. A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (miqcqp) through piecewise-linear and edge-concave relaxations. *Mathematical Programming Series B*, 136:155–182, 2012.
9. R. Misener and C. A. Floudas. GloMIQO: Global Mixed-Integer Quadratic Optimizer. *Journal of Global Optimization*, 2012. Accepted for Publication (DOI: 10.1007/s10898-012-9874-7).
10. P. M. Pardalos and J. B. Rosen. *Constrained Global Optimization: Algorithms and Applications*. Lecture Notes in Computer Science. Springer, 1987.
11. S. Rebennack and J. Kallrath. Continuous Piecewise Linear δ-Approximations for MINLP Problems. I. Minimal Breakpoint Systems for Univariate Functions. *Journal of Optimization Theory and Applications*, submitted, 2012.
12. S. Rebennack and J. Kallrath. Continuous Piecewise Linear δ-Approximations for MINLP Problems. II. Bivariate and Multivariate Functions. *Journal of Optimization Theory and Applications*, submitted, 2012.
13. J. B. Rosen and P. M. Pardalos. Global minimization of large-scale constrained concave quadratic problems by separable programming. *Mathematical Programming*, 34:163–174, 1986.
14. L. Schrage. LindoSystems: LindoAPI, 2004.
15. J. P. Vielma and G. Nemhauser. Modeling Disjunctive Constraints with a Logarithmic Number of Binary Variables and Constraints. *Mathematical Programming*, 128:49–72, 2011.