Piecewise Linear Function Fitting via Mixed-Integer Linear Programming

Steffen Rebennack*
Institute of Operations Research, Karlsruhe Institute of Technology, steffen.rebennack@kit.edu

Vitaliy Krasko
Division of Economics and Business, Colorado School of Mines

Piecewise linear (PWL) functions are used in a variety of applications. Computing such continuous PWL functions, however, is a challenging task. Software packages and the literature on PWL function fitting are dominated by heuristic methods. This is true for both fitting discrete data points and continuous univariate functions. The only exact methods rely on non-convex model formulations. Exact methods compute continuous PWL function for a fixed number of breakpoints minimizing some distance function between the original function and the PWL function. An optimal PWL function can only be computed if the breakpoints are allowed to be placed freely and are not fixed to a set of candidate breakpoints. In this paper, we propose the first convex model for optimal continuous univariate PWL function fitting. Dependent on the metrics chosen, the resulting formulations are either mixed-integer linear programming or mixed-integer quadratic programming problems. These models yield optimal continuous PWL functions for a set of discrete data. Based on these convex formulations, we further develop an exact algorithm to fit continuous univariate functions. Computational results for benchmark instances from the literature demonstrate the superiority of the proposed convex models compared to state-of-the-art non-convex models.

Key words: Piecewise linear function; linear spline; splines of degree 1; polyhedral function; mixed-integer linear programming (MILP); mixed-integer quadratic programming (MIQP); function fitting; spline regression; global optimization

1. Introduction

Fitting discrete data points with a continuous function is a common technique in various applications in which the true function is unknown. Although this is commonly done with polynomials, there are certain advantages to using piecewise linear (PWL) functions. PWL

* Corresponding author

functions are composed of affine functions. Points where these affine functions intersect are called the breakpoints of the PWL function.

If the resulting fitted function is to be minimized or maximized subject to a set of constraints, a computational advantage may be found by using a PWL representation and avoiding the inherent non-linearities of the polynomial. While fitting such a function is straightforward when the breakpoints or knots connecting the linear pieces are known, significant computational issues arise when we seek to determine the globally optimal locations for a fixed number of free breakpoints. Similarly, dropping the continuity requirement for the PWL function to be constructed makes the problem considerably easier (Chen and Wang 2009).

Similarly, continuous univariate functions may be approximated by continuous PWL functions. This is of particular practical importance when approximating non-linear functions. The goal is often to approximately solve (mixed-integer) non-linear and non-convex programming problems by (mixed-integer) linear programming techniques via the use of such continuous PWL functions (Geißler et al. 2012, Feijoo and Meyer 1988). Again, the computation of such PWL functions is a challenging task when the breakpoint locations are not fixed.

Piecewise linear functions appear in many different disciplines. They are also known as splines of degree 1 in approximation theory. In statistics, continuous PWL functions are called linear splines. Continuous PWL function fitting is also known as linear spline regression. In mathematics, PWL functions are also referred to as polyhedral functions.

The literature on PWL function representation has grown over the past decade (Vielma et al. 2010). There is now a whole suite of different models available, spanning for example the convex, multiple-choice (Croxton et al. 2003), disaggregated (Sherali 2001), incremental (Padberg 2000) or logarithmic formulations (Vielma and Nemhauser 2011). While there are some theoretical results on local idealness and the formulations differ in the number of decision variables and constraints, the best model still depends on the particular application at hand. These models have all in common that they assume the availability of a (continuous) PWL function (Rebennack 2016c). In contrast, in this paper, we are concerned with the computation of optimal continuous PWL functions. In this context, a PWL linear function is optimal when it minimizes the given distance function.

While the problem of PWL fitting is well studied in the literature, it is rarely approached from the perspective of global optimization. In one of the earliest papers on the subject, Bellman and Roth (1969) use a dynamic programming approach to minimize the maximum-norm by optimally selecting breakpoints from a uniform grid of candidate breakpoints. The assumption of the candidate breakpoints essentially limits the feasible region and the technique therefore does not guarantee a globally optimal solution to the original problem. Jupp (1978) provides a numerical approach to the problem and demonstrates the existence of stationary points on the boundary of the domain as well the existence of a large number of local optima that lead to the "lethargy" property. These issues have led to the development of various suboptimal but practical techniques including a greedy hierarchical scheme (McGee and Carleton 1970) and an efficient dynamic program that drops the continuity requirement (Guthery 1974). Various heuristics have also been proposed (Ertel and Fowlkes 1976, Magnani and Boyd 2009). The R package *segmented* implements an iterative procedure that converges at local optima (Muggeo 2003). The *segmented* package uses bootstrap restarting (Wood 2001) to make the algorithm less sensitive to initial breakpoint values, which are required. The need for a minimal number of segments to meet a given maximal deviation was already recognized by Williams (1978) while the proposed geometric algorithm cannot guarantee optimality of the computed PWL function.

Various MILP models have been proposed in the literature for function fitting. Among the first integer programming approaches are by Bertsimas and Shioda (2007). They use MILP models for classification and then fit discontinuous PWL functions. Classical regression problems from statistics are tackled via MILP approaches by Bertsimas and Mazumder (2014), Bertsimas and King (2016) and by Bertsimas et al. (2016). All these problems are related to PWL function fitting. However, they do not contain the complicating continuity requirement on the computed functions.

Goldberg et al. (2014) introduces a non-convex non-linear approach to approximating a discrete set of data points with a continuous PWL function. The non-convexity originates from the continuity requirement of the PWL function. A dynamic programming approach is introduced with an adaptive refinement scheme for the grid of candidate breakpoints. It is demonstrated that the dynamic programming approach has an advantage in terms of computation speed but there is often a large gap between the resulting solution and the global optimum as a consequence of its heuristic nature. Toriello and Vielma (2012) introduce a non-convex mixed-integer quadratically constrained programming problem that solves a similar fitting problem to global optimality using polytopes. This approach differs from Goldberg et al. (2014) because Goldberg et al. (2014) force the distance function between adjacent breakpoints to always correspond to the same line segment.

Furthermore, Toriello and Vielma (2012) introduces a MILP model to fit convex PWL functions to discrete data. Convex PWL functions are special in that any function value is given by the maximum among all linear segments. This maximum operator is then reformulated using Big-M constructs by Toriello and Vielma (2012). A similar reformulation is used in Bertsimas and Shioda (2007). We also make use of Big-M constraints, but in a different context and to yield general PWL functions.

One of the first approaches to approximate continuous univariate functions was presented by Hamann and Chen (1994). A set of breakpoints is computed taking the curvature of the continuous function into account. This yields a heuristic method. PWL interpolators with equidistant breakpoints for approximating concave quadratic programs are introduced by Rosen and Pardalos (1986). Geißler (2011) present some heuristic methods to fit continuous PWL functions. Rebennack and Kallrath (2015) introduce the first exact model to fitting continuous univariate functions; the proposed models are all non-linear and non-convex.

For fitting a set of discrete data, several different distance functions between the PWL function and the data points have been proposed. We discuss the minimization of (1) the sum of differences, of (2) the sum of squared differences and of (3) the maximum difference. For the continuous function fitting, we restrict our discussion to minimization of the maximum difference.

In this paper, we introduce the first convex[1] models for optimal continuous univariate PWL function fitting. Therefore, the key insight is to recognize that the exact location of

[1] While we recognize that the feasible region of mixed-integer (linear or non-linear) programming models are not convex sets, we call such models convex if the associated relaxation of the integrality constraints yield a convex model.

the breakpoints is not necessary during the computation of the PWL functions. Rather, it is sufficient to ensure that adjacent linear segments of the constructed PWL functions intersect within a certain range. Such a condition can be ensured via mixed-integer linear programming (MILP) techniques. For metrics (1) and (3), the resulting exact models are then MILP problems. For (2), we obtain convex mixed-integer quadratic programming (MIQP) problems.

Based on the convex models for discrete data, we construct an exact algorithm to fit continuous univariate functions. The idea is to solve a series of discrete data fitting problems via the proposed MILP models. By evaluation of the computed PWL function, we can identify points where the maximum difference between the PWL function and the original function is larger than desired. These points are dynamically added to the discretization and the discrete fitting problem is resolved. This way, a finitely convergent algorithm is constructed.

This paper has the following main contributions:

- For discrete data, we present MILP and convex MIQP models to compute globally optimal univariate PWL functions, for three different metrics. These are the first convex models to optimally compute PWL functions. When restricting the computations to *convex* PWL functions, the derived MILPs and convex MIQPs reduce to the model presented by Toriello and Vielma (2012).
- For univariate functions, we present a finitely convergent algorithm which computes globally optimal PWL functions minimizing the maximum difference. The main component of the algorithm are MILP models. Globally optimal solutions need only be computed for univariate, box-constrained problems. These models can be efficiently computed via existing software packages. The resulting algorithm is the first exact method to compute globally optimal PWL functions, based on MILP models.

The remainder is organized as follows. In Sect. 2, we revisit the state-of-the-art nonconvex model formulation for fitting a set of discrete data. Our new convex models are then presented in Sect. 3 to compute globally optimal continuous PWL functions for a set of discrete data. Sect. 4 extends these ideas to continuous univariate function fitting. We present our computational results for both discrete data and continuous function fitting in Sect. 5 before we conclude with Sect. 6.

2. State-of-the-Art Model Formulations for Discrete Data Fitting

In this section, we introduce the problem and describe the state-of-the-art model from the literature (Sect. 2.1). This model is chosen as a benchmark because it is the only global optimization model which does not require candidate breakpoints or assumptions about convexity. We discuss some important features including different distance functions or metrics (Sect. 2.2) and "Big-M" constants together with variable bounds (Sect. 2.3). We summarize the resulting models for the different metrics (Sect. 2.4) and prove their correctness (Sect. 2.5).

We start by formalizing continuous univariate PWL functions.

Definition 1. A continuous function $p(x): [\underline{X}, \overline{X}] \to \mathbb{R}$ with compact interval $[\underline{X}, \overline{X}]$ is called a continuous *piecewise linear function*, if there exists a finite number B with

$$\underline{X} = r_1 < \ldots < r_b < r_{b+1} < \ldots < r_B = \overline{X} \tag{1}$$

such that $p(x)$ is an affine function on $[r_b, r_{b+1}]$ for all $b = 1, \ldots, B-1$. The r_b are called *breakpoints* and B is the *number of breakpoints*. For each $b = 1, \ldots, B-1$, the function $p(x): [r_b, r_{b+1}] \to \mathbb{R}$ is called a *linear segment*. □

2.1. Non-convex Model Formulations for Univariate Functions

We are given a set of I data points $(X_i, Y_i) \in \mathbb{R}^2$, $i = 1, \ldots, I$, and we assume that

$$-\infty < \underline{X} = X_1 < \ldots < X_i < X_{i+1} < \ldots < X_I = \overline{X} < \infty, \tag{2}$$

i.e., (X_i, Y_i) are the sorted values of a discrete function from $[\underline{X}, \overline{X}]$ to $\mathbb{R}$. We seek a continuous PWL function $p(x): [\underline{X}, \overline{X}] \to \mathbb{R}$ which approximates these I data points for a given number of breakpoints B. Specifically, we want to compute a continuous PWL function $p(\cdot)$ which optimizes a given metric $d(\cdot,\cdot)$. Moreover, we aim at computing the best possible PWL together with an optimality guarantee, subject to some mild assumptions.

The distance function $d(\cdot,\cdot): \mathbb{R}^I \times \mathbb{R}^I \to [0, \infty)$ is a metric which measures the distance between the data points Y_i and the functions values $y_i = p(X_i)$ for $i = 1, \ldots, I$.

For the construction of an optimal PWL function, it is necessary to allow the breakpoints to be free within the range $[\underline{X}, \overline{X}]$. In the formulation below, they enter the optimization problem as decision variables r_b, $b=1,\dots,B$. (In our proposed formulation in Sect. 3, they are implicitly given.) A PWL function with B breakpoints has $B-1$ linear segments, $b=1,\dots,B-1$. Each linear segment is defined by its slope c_b, intercept d_b, and the interval $[r_b, r_{b+1}]$.

Consider the following non-convex mathematical programming formulation based on Goldberg et al. (2014):

$$\min\ d\big((Y_1,Y_2,\dots,Y_I)^\top,(y_1,y_2,\dots,y_I)^\top\big) \tag{3a}$$

$$\text{s.t. } c_bX_i + d_b \le y_i + M_i^1(1-\delta_{i,b}) \qquad \forall b=1,\dots,B-1,\ i=1,\dots,I \tag{3b}$$

$$c_bX_i + d_b \ge y_i - M_i^1(1-\delta_{i,b}) \qquad \forall b=1,\dots,B-1,\ i=1,\dots,I \tag{3c}$$

$$\sum_{b=1}^{B-1}\delta_{i,b} = 1 \qquad \forall i=1,\dots,I \tag{3d}$$

$$-M_i^2(1-\delta_{i,b}) \le r_{b+1} - X_i \qquad \forall b=1,\dots,B-2,\ i=1,\dots,I \tag{3e}$$

$$-M_i^3(1-\delta_{i,b}) \le X_i - r_b \qquad \forall b=2,\dots,B-1,\ i=1,\dots,I \tag{3f}$$

$$c_b r_{b+1} + d_b = c_{b+1}r_{b+1} + d_{b+1} \qquad \forall b=1,\dots,B-2 \tag{3g}$$

$$r_b \le r_{b+1} \qquad \forall b=1,\dots,B-1 \tag{3h}$$

$$y_i \in [\underline{M}_i^4, \overline{M}_i^4] \qquad \forall i=1,\dots,I \tag{3i}$$

$$d_b \in [\underline{D}_b, \overline{D}_b] \qquad \forall b=1,\dots,B-1 \tag{3j}$$

$$c_b \in [\underline{C}_b, \overline{C}_b] \qquad \forall b=1,\dots,B-1 \tag{3k}$$

$$\delta_{i,b} \text{ binary} \qquad \forall b=1,\dots,B-1,\ i=1,\dots,I \tag{3l}$$

$$r_b \in [\underline{X}, \overline{X}] \qquad \forall b=2,\dots,B-1 \tag{3m}$$

$$r_1 = X_1,\ r_B = X_I. \tag{3n}$$

Objective function (3a) minimizes the distance function of data points Y_i to the PWL function evaluated at X_i for $i=1,\dots,I$. We discuss three different metrics in Sect. 2.2. Constraint group (3b)-(3c) evaluate the PWL function at the points X_i, *i.e.*, $y_i = p(X_i)$ for $i=1,\dots,I$. Therefore, binary variable $\delta_{i,b}$ equals 1, iff data point (X_i, Y_i) is associated with segment b of the PWL function ($i=1,\dots,I, b=1,\dots,B-1$). Decision variables c_b and d_b are the slope and the intercept, respectively, of the linear segment b of the PWL function

($b=1,\ldots,B-1$). All Big-M constants are discussed in Sect. 2.3. That each data point is associated with exactly one linear segment is enforced by (3d). The group (3e)-(3f) ensures that data point X_i lies between the two endpoints of the assigned linear segment b, *i.e.*, $X_i \in [r_b, r_{b+1}]$. Constraints (3g) enforce continuity by forcing adjacent linear segments to be equal when evaluated at their shared breakpoint. The bilinear terms $c_b r_{b+1}$ and $c_{b+1} r_{b+1}$ in constraints (3g) yield a non-convex feasible region. The ordering of the breakpoints is enforced explicitly through constraints (3h); this constraint presents the only difference to the formulation in Goldberg et al. (2014). The remaining constraints (3j)-(3n) define the domain of the decision variables. The decision variable bounds for (3i)-(3k) are discussed in Sect. 2.3.

2.2. Three Different Metrics

There are three different distance functions that are commonly minimized in the discrete fitting problem: the sum of absolute differences, the sum of squared differences, and the maximum difference.

The sum of absolute differences, the ℓ_1-norm, can be modeled by

$$d\big((Y_1,Y_2,\ldots,Y_I)^\top,(y_1,y_2,\ldots,y_I)^\top\big)=\sum_{i=1}^{I}\xi_i \tag{4a}$$

$$\text{s.t. } Y_i-\big(c_bX_i+d_b\big)\le \xi_i+M_i^{\mathrm{a}}(1-\delta_{i,b}) \qquad \forall b=1,\ldots,B-1,\ i=1,\ldots,I \tag{4b}$$

$$\big(c_bX_i+d_b\big)-Y_i\le \xi_i+M_i^{\mathrm{a}}(1-\delta_{i,b}) \qquad \forall b=1,\ldots,B-1,\ i=1,\ldots,I \tag{4c}$$

$$\xi_i\in[\underline{M}_i^{\xi},\overline{M}_i^{\xi}] \qquad \forall i=1,\ldots,I \tag{4d}$$

Constraints (4b)-(4c) define the value of variables ξ_i as the absolute difference of Y_i and its associated segment's b function value $p(X_i)$, *i.e.*,

$$\xi_i=\sum_{b=1}^{B-1}\big|Y_i-\big(c_bX_i+d_b\big)\big|\cdot\delta_{i,b} \qquad \forall i=1,\ldots,I. \tag{5}$$

The sum of the absolute differences is then given by (4a). The Big-M value and the domain (4d) are discussed below. Notice that (4b)-(4c) are linear constraints, in contrast to (5) which contains a non-differentiable function on the right-hand-side (RHS).

The sum of squared differences, the ℓ_2-norm squared, is simply given by

$$d\big((Y_1,Y_2,\ldots,Y_I)^\top,(y_1,y_2,\ldots,y_I)^\top\big)=\sum_{i=1}^{I}\xi_i^2 \tag{6a}$$

$$\text{s.t. } (4b)-(4d). \tag{6b}$$

Minimizing (6a) yields a convex, quadratic objective function. The sum of squared differences weighs the outliers more heavily compared to the sum of absolute differences.

The maximum difference, the ℓ_∞-norm, can be formulated by removing the index i from decision variables ξ_i and leaving the rest untouched, compared to the sum of absolute differences.

$$d\big((Y_1,Y_2,\ldots,Y_I)^\top,(y_1,y_2,\ldots,y_I)^\top\big)=\xi \tag{7a}$$

$$\text{s.t. } Y_i-\big(c_bX_i+d_b\big)\le\xi+M_i^{\text{a}}(1-\delta_{i,b}) \qquad \forall b=1,\ldots,B-1,\ i=1,\ldots,I \tag{7b}$$

$$\big(c_bX_i+d_b\big)-Y_i\le\xi+M_i^{\text{a}}(1-\delta_{i,b}) \qquad \forall b=1,\ldots,B-1,\ i=1,\ldots,I \tag{7c}$$

$$\xi\in[\underline{M}^\xi,\overline{M}^\xi]. \tag{7d}$$

The maximum difference essentially brings the approximated function (*i.e.*, the constructed PWL function) as close as possible to the farthest resulting outlier, without considering the remaining differences to the other data points.

2.3. Variable Domain and Big-M: Deriving the Constants

Formulation (3a)-(3n), with the three different metrics, requires numeric values for the sufficiently large constants M_i^1, M_i^2, M_i^3, M_i^{a} and variable bounds $\underline{M}_i^4$, $\overline{M}_i^4$, $\underline{M}_i^\xi$, $\overline{M}_i^\xi$, $\underline{M}^\xi$ and $\overline{M}^\xi$.

The first constant should be great than or equal to the largest function value at the data points X_i. For given bounds on the slope and intercept, we can set

$$M_i^1=\max\left\{\,\big|\overline{C}X_i+\overline{D}\big|\,,\ \big|\overline{C}X_i+\underline{D}\big|\,,\big|\underline{C}X_i+\overline{D}\big|\,,\ \big|\underline{C}X_i+\underline{D}\big|\,\right\} \qquad \forall i=1,\ldots,I.$$

From (3e)-(3f) we derive

$$M_i^2=X_i-\underline{X} \qquad \text{and} \qquad M_i^3=\overline{X}-X_i \qquad \forall i=1,\ldots,I.$$

Constants M_i^{a} should be greater than or equal to the largest possible difference between the data point Y_i and any of the linear functions fit, *i.e.*,

$$M_i^{\mathrm{a}} \geq \max_{b=1,\ldots,B-1} |Y_i - (c_b X_i + d_b)| \qquad \forall i = 1,\ldots,I,$$

so we can set

$$\begin{aligned} M_i^{\mathrm{a}} = \max\Big\{ & \left|Y_i - \overline{C}X_i - \overline{D}\right|, \left|Y_i - \overline{C}X_i - \underline{D}\right|, \\ & \left|Y_i - \underline{C}X_i - \overline{D}\right|, \left|Y_i - \underline{C}X_i - \underline{D}\right| \Big\} \qquad \forall i = 1,\ldots,I. \end{aligned}$$

For the variable bounds, we obtain

$$\begin{aligned} \underline{M}_i^4 &= -\overline{M}_i^4 = -M_i^1 && \forall i = 1,\ldots,I \\ \underline{M}_i^{\xi} &= -\overline{M}_i^{\xi} = -M_i^{\mathrm{a}} && \forall i = 1,\ldots,I \\ \underline{M}^{\xi} &= -\overline{M}^{\xi} = -\max_{i=1,\ldots,I} M_i^{\mathrm{a}}. \end{aligned}$$

So far, we assume the availability of bounds on the slopes and intercepts. Although it may not be straightforward how to bound the slopes and intercepts, in practice it is necessary to define these bounds in order to solve the non-convex non-linear problem to global optimality. The assumption of slope bounds can be justified because we do not want an infinite slope or an arbitrarily large one. Because these bounds affect the optimal solution computed as well as the computational times, they need to be derived consistently.

A natural choice for the bounds on the slope are the extreme slopes obtained by interpolating any two data points which are shifted up and down by some value $\kappa \geq 0$, *i.e.*,

$$\underline{C} = \min_{i,j:i>j} \left\{ \frac{Y_i - Y_j}{X_i - X_j} - \kappa \left| \frac{Y_i - Y_j}{X_i - X_j} \right| \right\} \quad \text{and} \quad \overline{C} = \max_{i,j:i>j} \left\{ \frac{Y_i - Y_j}{X_i - X_j} + \kappa \left| \frac{Y_i - Y_j}{X_i - X_j} \right| \right\}. \tag{8}$$

In Goldberg et al. (2014) the authors choose $\kappa = 0$ and then bound the intercept by finding the minimum and maximum intercepts among the linear functions constructed by interpolating of every set of two points. A more general approach is to find the extreme values for the intercepts through an affine function that has an extreme slope and passes through some "extreme point", *i.e.*,

$$\underline{D} = \min_i \left\{ Y_i - \overline{C}X_i,\, Y_i - \underline{C}X_i \right\} \quad \text{and} \quad \overline{D} = \max_i \left\{ Y_i - \overline{C}X_i,\, Y_i - \underline{C}X_i \right\}. \tag{9}$$

2.4. Complete Non-convex Models

A summary of the resulting non-convex model formulations are given in Table 1. We name the three non-convex fitting problem formulations for the different metrics $\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$, $\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$ and $\mathcal{F}^{\max}_{\text{MIQCP}}(B)$. The sum of absolute differences and the maximum difference yield mixed-integer quadratically-constrained programming (MIQCP) problems. Because the sum of squared differences contain a quadratic objective function, it belongs to the class of mixed-integer quadratically-constrained quadratic programming (MIQCQP) problems.

Table 1 **Different non-convex model formulations.**

Metric	**Sum of Absolute Differences**	**Sum of Squared Differences**	**Maximum Difference**
Model name	$\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$	$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$
Constraints	(3d)-(3n), (4a)-(4d)	(3d)-(3n), (4b)-(4d), (6a)	(3d)-(3n), (7a)-(7d)
Cont. Variables	c_b, d_b, r_b, ξ_i	c_b, d_b, r_b, ξ_i	c_b, d_b, r_b, ξ
# Cont. Variables	$3B + I - 4$	$3B + I - 4$	$3B - 3$
Binary Variables	$\delta_{i,b}$	$\delta_{i,b}$	$\delta_{i,b}$
# Binary Variables	$I(B-1)$	$I(B-1)$	$I(B-1)$
# Functional Constr.	$4IB + 2B - 5I - 3$	$4IB + 2B - 5I - 3$	$4IB + 2B - 5I - 3$
# Non-convex Terms	$2B - 4$	$2B - 4$	$2B - 4$
Model Type	non-convex MIQCP	non-convex MIQCQP	non-convex MIQCP

2.5. Correctness

By construction of the three non-convex formulations $\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$, $\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$ and $\mathcal{F}^{\max}_{\text{MIQCP}}(B)$, any resulting solution is a continuous PWL function. The optimality with respect to the three metrics is also guaranteed. For that, however, we need to show that there exists an optimal continuous PWL function which does not contain more than one breakpoint between any two consecutive data points. We assume $B \leq I$. (This is not a restriction for practical applications because with $B \geq I$ a perfect fit can always be reached.)

Let there be an optimal PWL function p^+, whose slopes and intercepts are bounded by $[\underline{C}, \overline{C}]$ and $[\underline{D}, \overline{D}]$, respectively, which cannot be modeled by the three non-convex formulations $\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$, $\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$ and $\mathcal{F}^{\max}_{\text{MIQCP}}(B)$. For example, these are functions with more than one breakpoint between the two data points at X_i and X_{i+1}, where the breakpoints do not equal the data points. We can construct a new continuous PWL function with the same objective function value having less (or the same number of) breakpoints by defining a breakpoint at each data point. In this segment, the PWL function is then defined via interpolation. The resulting affine function meets the slope and intercept bounds because p^+ is continuous!

Finally, the correctness of the non-convex models is established in the next

Theorem 1. *Given a set of I data points $(X_i, Y_i) \in \mathbb{R}^2$, $i = 1, \ldots, I$, satisfying* (2), *bounds on slope $[\underline{C}, \overline{C}]$ and intercept $[\underline{D}, \overline{D}]$ as well as number of breakpoints $B \leq I$. Optimal solutions of the three formulations $\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$, $\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$ and $\mathcal{F}^{\max}_{\text{MIQCP}}(B)$, as defined in Table 2.4, define (globally optimal) continuous PWL functions, minimizing the corresponding distance function, with slope $\in [\underline{C}, \overline{C}]$ and intercept $\in [\underline{D}, \overline{D}]$.* □

3. Novel MILP and Convex MIQP Formulations for Discrete Data Fitting

In this section, we introduce an exact MILP formulation for fitting continuous PWL functions to discrete data for the two metrics sum of absolute differences and maximum difference. When minimizing the sum of squared errors, the resulting formulation is a convex mixed-integer quadratic programming (MIQP) problem. This is the first globally optimal *linear (or convex)* formulation for fitting continuous piecewise linear functions (with free breakpoints) to discrete data. The key constraint that makes the original formulation nonlinear is continuity constraint (3g). Allowing the linear segments to have discontinuities would reduce the problem to a MILP, however, this leads to a different (much easier) fitting problem.

We start by discussing the fit of general PWL functions in Sect. 3.1. The special case of fitting a convex PWL function is then discussed in Sect. 3.2, where we reduce our models to a known model from the literature, which is tailored towards fitting convex PWL functions.

3.1. Fitting General Piecewise Linear Functions to Discrete Data

In order to make the problem linear, we reformulate the continuity constraint (3g) in a way that does not require an explicit breakpoint decision variable r_b. The logic behind the new formulation is that we need to ensure adjacent linear segments intersect in an appropriate range. As long as this happens, we do not need to know the exact location of the intersection. The value of the breakpoint, r_b, can then be computed from the information of the two adjacent segments' slopes and intercepts. The appropriate range for this intersection would be between the x-coordinates of the last data point approximated by segment b and the first data point approximated by segment $b+1$. As long as the x-coordinate of the intersection is somewhere between these two data points, the continuity requirement is met. This is illustrated in Fig. 1 which shows part of the Titanium data set with the optimal solution. The breakpoints ranges are determined by the last and first data points associated with the corresponding line segments, as shown in the figure.

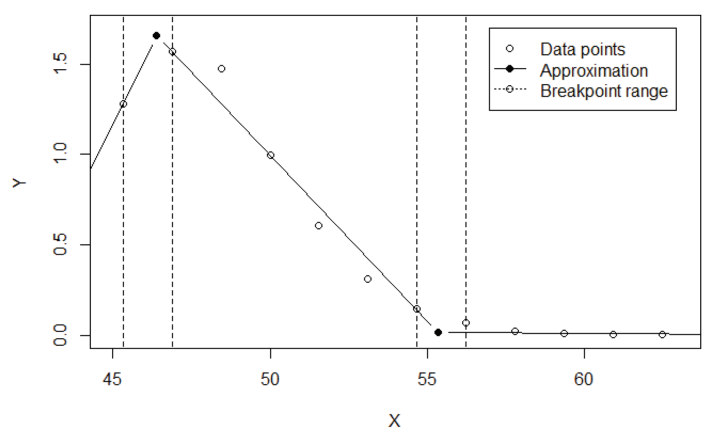


Figure 1 Partial Titanium data set and solution with breakpoint range for enforcing continuity.

Consider some breakpoint $\hat{r} \in [\underline{X}, \overline{X}]$ that connects linear segments $\hat{b}$ and $\hat{b}+1$ for some $\hat{b} \in \{1, \dots, B-2\}$. Starting with the continuity constraint (3g), we can move some terms around to express $\hat{r}$ as a function of the slopes and intercepts of the two associated line segments (without loss of generality assuming $c_{\hat{b}} \neq c_{\hat{b}+1}$, we come back to this assumption later):

$$c_{\hat{b}}\hat{r} + d_{\hat{b}} = c_{\hat{b}+1}\hat{r} + d_{\hat{b}+1} \qquad \Longleftrightarrow \qquad \hat{r} = \frac{d_{\hat{b}+1} - d_{\hat{b}}}{c_{\hat{b}} - c_{\hat{b}+1}}. \tag{10}$$

Now, because the breakpoint $\hat{r} \in [\underline{X}, \overline{X}]$ and because the data points partition the domain $[\underline{X}, \overline{X}]$, there exists some $\hat{\imath} \in \{1, \ldots, I\}$ such that $\hat{r} \in [X_{\hat{\imath}}, X_{\hat{\imath}+1}]$, *i.e.*, any breakpoint must lie between the x-coordinates of two adjacent data points, or can be equal to one of them. Thus, if continuity (10) holds, then

$$X_{\hat{\imath}} \leq \frac{d_{\hat{b}+1} - d_{\hat{b}}}{c_{\hat{b}} - c_{\hat{b}+1}} \leq X_{\hat{\imath}+1}. \tag{11}$$

Also, (11) implies (10). Observe that (11) only applies for the pair $(\hat{\imath}, \hat{b})$ if $\delta_{\hat{\imath},\hat{b}} = \delta_{\hat{\imath}+1,\hat{b}+1} = 1$.

Next, we distribute the denominator in (11) to avoid non-linearities. In order to do so, we need to know whether the denominator is positive or negative as this determines the direction of the inequalities. Let γ_b be a binary decision variable that can be equal to one if $c_b - c_{b+1} \geq 0$ and zero if $c_b - c_{b+1} \leq 0$. This can be enforced by the following constraints:

$$c_b - c_{b+1} \geq -M^5 (1 - \gamma_b) \qquad \forall b = 1, \ldots, B-2 \tag{12a}$$

$$c_b - c_{b+1} \leq M^5 \gamma_b \qquad \forall b = 1, \ldots, B-2 \tag{12b}$$

with the Big-M constant $M^5 \geq \overline{C} - \underline{C}$.

We obtain the following convex model for the continuity constraints (3g), $\forall i = 1, \ldots, I - 1;\ b = 1, \ldots, B-2$

$$d_{b+1} - d_b \geq X_i (c_b - c_{b+1}) - M^6_{i,b}\big((1 - \delta_{i,b}) + (1 - \delta_{i+1,b+1}) + (1 - \gamma_b)\big) \tag{13a}$$

$$d_{b+1} - d_b \leq X_{i+1} (c_b - c_{b+1}) + M^6_{i,b}\big((1 - \delta_{i,b}) + (1 - \delta_{i+1,b+1}) + (1 - \gamma_b)\big) \tag{13b}$$

$$d_{b+1} - d_b \leq X_i (c_b - c_{b+1}) + M^6_{i,b}\big((1 - \delta_{i,b}) + (1 - \delta_{i+1,b+1}) + \gamma_b\big) \tag{13c}$$

$$d_{b+1} - d_b \geq X_{i+1} (c_b - c_{b+1}) - M^6_{i,b}\big((1 - \delta_{i,b}) + (1 - \delta_{i+1,b+1}) + \gamma_b\big) \tag{13d}$$

In constraints (13a)-(13d), the Big-M constants $M^6_{i,b}$ are multiplied by the sum of three variable terms. Each constraint is activated, if all three decision variable terms are equal to zero. Constraints (13a)-(13d) can be strengthened by replacing the three terms, associated with the Big-M constants $M^6_{i,b}$, by a single variable having range [0,1]; we explain this below.

Lastly we can use the special structure of the binary $\delta_{i,b}$ variables to find an alternative formulation of constraints (3h):

$$\delta_{i+1,b+1} \le \delta_{i,b} + \delta_{i,b+1} \qquad \forall i = i,\dots,I-1;\ b = 1,\dots,B-2 \tag{14a}$$

$$\delta_{i+1,1} \le \delta_{i,1} \qquad \forall i = 1,\dots,I-1 \tag{14b}$$

$$\delta_{i,B-1} \le \delta_{i+1,B-1} \qquad \forall i = 1,\dots,I-1. \tag{14c}$$

Constraint (14a) states that in order for point $i+1$ to be associated with line segment $b+1$, the left adjacent point i must either be associated with the same segment or the previous segment b. Constraints (14b) and (14c) essentially apply the same logic to the first and last linear segments. It states that in order for data point $i+1$ to be associated with segment $b=1$, the left adjacent point i must also be associated with that segment and similarly for $b=B-1$ and the right adjacent point.

We obtain the following convex formulation for fitting discrete data:

$$\min\ d\big((Y_1,Y_2,\dots,Y_I)^\top,(y_1,y_2,\dots,y_I)^\top\big) \tag{15a}$$

$$\text{s.t. } (3\text{b})-(3\text{d}),(14\text{a})-(14\text{c}) \tag{15b}$$

$$\delta_{i,b} + \delta_{i+1,b+1} + \gamma_b - 2 \le \delta^+_{i,b} \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2 \tag{15c}$$

$$\delta_{i,b} + \delta_{i+1,b+1} + (1-\gamma_b) - 2 \le \delta^-_{i,b} \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2 \tag{15d}$$

$$d_{b+1} - d_b \ge X_i(c_b - c_{b+1}) - M^7_i(1-\delta^+_{i,b}) \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2 \tag{15e}$$

$$d_{b+1} - d_b \le X_{i+1}(c_b - c_{b+1}) + M^7_{i+1}(1-\delta^+_{i,b}) \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2 \tag{15f}$$

$$d_{b+1} - d_b \le X_i(c_b - c_{b+1}) + M^7_i(1-\delta^-_{i,b}) \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2 \tag{15g}$$

$$d_{b+1} - d_b \ge X_{i+1}(c_b - c_{b+1}) - M^7_{i+1}(1-\delta^-_{i,b}) \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2 \tag{15h}$$

$$y_i \in [\underline{M}^4_i, \overline{M}^4_i] \qquad \forall i = 1,\dots,I \tag{15i}$$

$$d_b \in [\underline{D}_b, \overline{D}_b] \qquad \forall b = 1,\dots,B-1 \tag{15j}$$

$$c_b \in [\underline{C}_b, \overline{C}_b] \qquad \forall b = 1,\dots,B-1 \tag{15k}$$

$$\delta_{i,b} \text{ binary} \qquad \forall i=1,\dots,I,\ b=1,\dots,B-1 \quad (15l)$$

$$\gamma_b \text{ binary} \qquad \forall b=1,\dots,B-2 \quad (15m)$$

$$\delta^+_{i,b}, \delta^-_{i,b} \in [0,1] \qquad \forall i=1,\dots,I-1,\ b=1,\dots,B-2 \quad (15n)$$

The objective function (15a) minimizes the chosen distance function. Together with the y_i values of the PWL function modeled in (3b)-(3c), they are taken from the non-convex model. Constraint (3d) also remains the same ensuring that each data point is associated with exactly one line segment. The breakpoint sorting is indirectly achieved by (14a)-(14c).

Constraint group (15c)-(15h) enforce continuity of the PWL segments. They are a strengthened formulation of (12a)-(12b) and (13a)-(13d). Recall that there is a breakpoint between data point X_i and X_{i+1} if and only if $\delta_{i,b} = \delta_{i+1,b+1} = 1$. In this case, either $\delta^+_{i,b}$ or $\delta^-_{i,b}$ is forced to 1 via (15c)-(15d). If $\delta^+_{i,b} = 1$, then constraints (15e)-(15f) are active. They become (11). The slopes between the two segments b and $b+1$ are decreasing, *i.e.*, $c_b - c_{b+1} \geq 0$. Similarly, for $\delta^-_{i,b} = 1$, constraints (15g)-(15h) are active to ensure (11) for the case of increasing slopes. Note that variables $\delta^+_{i,b}$ and $\delta^-_{i,b}$ are not required to be binary because of constraints (15c)-(15d). For the Big-M constants holds

$$M^7_i \geq \overline{D} - \underline{D} - X_i \left(\underline{C} - \overline{C}\right) \qquad \forall i=1,\dots,I;\ b=1,\dots,B-2.$$

Constraints (15i)-(15n) summarize the decision variables with their domain.

Given the structure (14a)-(14c), it is safe to assume that $\delta_{1,1} = \delta_{I,B-1} = 1$. In other words, we know that the first data point is associated with the first linear segment while the last data point is associated with the last linear segment. We can extend this to the other data points as well since each segment must have at least one data point; *i.e.*, $\delta_{i,b} = 0$ where $b > i$ or $b < (B-1-I+i)$. This results in an insignificant problem size reduction.

Now, let c^*_b, d^*_b and $\delta^*_{i,b}$ define an optimal solution to (15a)-(15n) for some appropriate distance function. The optimal breakpoints can then be calculated by

$$\begin{aligned}
r^*_1 &= X_1 \\
r^*_b &= \begin{cases} \dfrac{d^*_{b+1} - d^*_b}{c^*_b - c^*_{b+1}}, & \text{if } c^*_b - c^*_{b+1} \neq 0 \\ X_i + \dfrac{X_{i+1} - X_i}{2}, & \text{o/w for } i \text{ with } \delta^*_{i,b} = \delta^*_{i+1,b+1} = 1 \end{cases} \qquad \forall b = 2,\dots,B-1 \\
r^*_B &= X_I
\end{aligned}$$

Note that $c_b - c_{b+1} = 0$ is allowed in formulation (15b)-(15n). In this case, $d_{b+1} - d_b = 0$ is enforced, *i.e.*, the two line segments lead to an affine function (there is no discontinuity of the PWL function in $[X_{i-1}, X_{i+1}]$). We arbitrarily define the corresponding breakpoint in the middle of the two associated data points.

We summarize the resulting linear and convex formulations for discrete data fitting in Table 2. Compared to the non-convex formulations (*cf.* Table 1), there are $B-2$ additional binary variables (γ_b) and $(B-2)(I-2)$ additional continuous variables ($\delta^+_{i,b}, \delta^-_{i,b}$ are in addition, r_b drops). Because (3e)-(3f) drop and (3h) get replaced by the constraint group (14a)-(14c), there are $5(B-2)(I-1)+2I-3B+3$ additional linear constraints in the three convex formulations. Most significantly, the $2B-4$ non-convex constraints are dropped in formulations $\mathcal{F}^{\text{abs}}_{\text{MILP}}(B)$, $\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$ and $\mathcal{F}^{\max}_{\text{MILP}}(B)$.

Because for practical applications, we typically have $B \ll I$, the increase in binary decision variables of $\mathcal{O}(B)$ is modest. However, there are $\mathcal{O}(BI)$ additional continuous variables and linear constraints.

Table 2 **Different convex model formulations.**

Metric	**Sum of Absolute Differences**	**Sum of Squared Differences**	**Maximum Difference**
Model name	$\mathcal{F}^{\text{abs}}_{\text{MILP}}(B)$	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	$\mathcal{F}^{\max}_{\text{MILP}}(B)$
Constraints	(3d), (4a)-(4d), (14a)-(14c), (15c)-(15n)	(3d), (4b)-(4d), (6a), (14a)-(14c), (15c)-(15n)	(3d), (7a)-(7d), (14a)-(14c), (15c)-(15n)
Cont. Variables	$c_b, d_b, \delta^+_{i,b}, \delta^-_{i,b}, \xi_i$	$c_b, d_b, \delta^+_{i,b}, \delta^-_{i,b}, \xi_i$	$c_b, d_b, \delta^+_{i,b}, \delta^-_{i,b}, \xi$
# Cont. Variables	$B(I+1)-I$	$B(I+1)-I$	$B(I+1)-2I+1$
Binary Variables	$\delta_{i,b}, \gamma_b$	$\delta_{i,b}, \gamma_b$	$\delta_{i,b}, \gamma_b$
# Binary Variables	$B(I+1)-I-2$	$B(I+1)-I-2$	$B(I+1)-I-2$
# Functional Constr.	$9IB-7B-13I+12$	$9IB-7B-13I+12$	$9IB-7B-13I+12$
# Non-convex Terms	0	0	0
Model Type	(convex) MILP	convex MIQP	(convex) MILP

By construction, the three convex formulations are equivalent to the corresponding three non-convex formulations. Thus, Theorem 1 implies

Theorem 2. *Given a set of I data points $(X_i, Y_i) \in \mathbb{R}^2$, $i = 1, \ldots, I$, satisfying (2), bounds on slope $[\underline{C}, \overline{C}]$ and intercept $[\underline{D}, \overline{D}]$ as well as number of breakpoints $B \leq I$. Optimal solutions of the three formulations $\mathcal{F}^{\text{abs}}_{\text{MILP}}(B)$, $\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$ and $\mathcal{F}^{\text{max}}_{\text{MILP}}(B)$, as defined in Table 3.1, define (globally optimal) continuous PWL functions, minimizing the corresponding distance function, with slope $\in [\underline{C}, \overline{C}]$ and intercept $\in [\underline{D}, \overline{D}]$.* □

The convex models of Table 2 avoid the $2B-4$ non-convex terms of the models in Table 1. However, this comes at the cost of an increased number of (linear) constraints and (binary and continuous) decision variables. More precisely, the $2(B-2)$ bilinear terms are replaced by $6(I-1)(B-1)$ linear constraints. Although $2IB+B$ constraints are not present in the convex models, these are less dense than the ones introduced. In addition, the number of non-linearities does not grow with the number of data points I in the non-convex models, while the more dense linear constraints in the convex models do.

Therefore, we can utilize a cutting plane approach where we iteratively add constraints of the group (15c)-(15h) rather than including all of them. Any computed (optimal) solution for the auxiliary problem formulation can be checked for correctness by inspecting the location of the breakpoints. If the PWL function is not continuous, or a breakpoint does not lie between the corresponding two data points, the violating constraints are added. This procedure terminates with an optimal continuous PWL function after finitely many steps, as there are only finitely many constraints (15c)-(15h).

3.2. Fitting Convex Piecewise Linear Functions to Discrete Data

A special case occurs, when the PWL function $p(x)$ to be fit is required to be convex. Next, we study how the convex model (15a)-(15n) can be adjusted to convex function fitting.

Simply adding the constraint

$$c_b \leq c_{b+1} \qquad \forall b = 1, \ldots, B-2 \tag{16}$$

to the model (15a)-(15n) ensures convexity of the computed PWL function $p(x)$.

However, the special structure, implied by a convex PWL function, has implications on model (15a)-(15n) & (16). Several constraints are redundant and the γ_b decision variables are fixed to 0. We detail this below.

Recall that decision variable y_i models the function value of the computed PWL function at the x-coordinate of data point i, *i.e.*, $y_i = p(X_i)$. Because of the convexity of $p(x)$,

$$y_i = \max_{b=1,\dots,B-1} c_b X_i + d_b \qquad \forall i = 1,\dots,I$$
$$\Longrightarrow y_i \geq c_b X_i + d_b \qquad \forall b = 1,\dots,B-1,\ i = 1,\dots,I,$$

a well known property which is commonly exploited (Rebennack 2016a, Lohmann and Rebennack 2017). Therefore, (3b) gets strengthend to

$$c_b X_i + d_b \leq y_i \qquad \forall b = 1,\dots,B-1,\ i = 1,\dots,I, \tag{17}$$

i.e., the term "$+M_i^1(1-\delta_{i,b})$" has been dropped.

With (3c) and (17) in place, constraints (14a)-(14c) are redundant. This can be seen as follows. Assume that $\delta_{i+1,b+1} = 1$ and $\delta_{i,b} = \delta_{i,b+1} = 0$. Because of the convexity of $p(x)$, data point i needs to assigned to a segment with index $< b$. This implies that segment b is not contained in $p(x)$, *i.e.*, $p(x)$ is constructed with less than B breakpoints. Because we allow $c_b = c_{b+1}$, there exists an alternate optimal solution with $\delta_{i,b} = 1$.

Because of (16), the decision variables $\gamma_b = 0$ for all $b = 1,\dots,B-2$. This implies that constraints (15c), (15e) and (15f) are always satisfied and can be eliminated along with decision variables $\delta^+_{i,b}$. Next, consider the remaining constraint block

$$\delta_{i,b} + \delta_{i+1,b+1} - 1 \leq \delta^-_{i,b} \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2 \tag{18a}$$
$$d_{b+1} - d_b \leq X_i(c_b - c_{b+1}) + M_i^7(1-\delta^-_{i,b}) \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2 \tag{18b}$$
$$d_{b+1} - d_b \geq X_{i+1}(c_b - c_{b+1}) - M_{i+1}^7(1-\delta^-_{i,b}) \qquad \forall i = 1,\dots,I-1;\ b = 1,\dots,B-2. \tag{18c}$$

The breakpoint enforcing constraints (18b)-(18c) are active, when $\delta_{i,b} = \delta_{i+1,b+1} = 1$. Therefore, let $\delta_{i,b} = \delta_{i+1,b+1} = 1$ which implies, by convexity of $p(x)$,

$$p(X_i) = c_b X_i + d_b \geq c_{b+1} X_i + d_{b+1} \qquad \Longrightarrow \qquad (c_b - c_{b+1})X_i \geq d_{b+1} - d_b$$
$$p(X_{i+1}) = c_{b+1} X_{i+1} + d_{b+1} \geq c_b X_{i+1} + d_b \qquad \Longrightarrow \qquad (c_{b+1} - c_b)X_i \geq d_b - d_{b+1}.$$

This implies that (18b)-(18c) are redundant as well; (18a) are no longer required and decision variables $\delta^-_{i,b}$ can be eliminated from the model.

In sum, we obtain the following convex formulation for fitting a convex function to discrete data (we re-state all constraints explicitly to enhance readability):

$$\min \; d\big((Y_1, Y_2, \dots, Y_I)^\top, (y_1, y_2, \dots, y_I)^\top\big) \tag{19a}$$

$$\text{s.t. } y_i \geq c_b X_i + d_b \qquad \forall b = 1, \dots, B-1,\; i = 1, \dots, I \tag{19b}$$

$$y_i \leq c_b X_i + d_b + M_i^1 (1 - \delta_{i,b}) \qquad \forall b = 1, \dots, B-1,\; i = 1, \dots, I \tag{19c}$$

$$\sum_{b=1}^{B-1} \delta_{i,b} = 1 \qquad \forall i = 1, \dots, I \tag{19d}$$

$$c_b \leq c_{b+1} \qquad \forall b = 1, \dots, B-2 \tag{19e}$$

$$y_i \in [\underline{M}_i^4, \overline{M}_i^4] \qquad \forall i = 1, \dots, I \tag{19f}$$

$$d_b \in [\underline{D}_b, \overline{D}_b] \qquad \forall b = 1, \dots, B-1 \tag{19g}$$

$$c_b \in [\underline{C}_b, \overline{C}_b] \qquad \forall b = 1, \dots, B-1 \tag{19h}$$

$$\delta_{i,b} \text{ binary} \qquad \forall i = 1, \dots, I,\; b = 1, \dots, B-1. \tag{19i}$$

Formulation (19a)-(19i) is identical to the model (3.12a)-(3.12g) together with the symmetry breaking constraints (3.13) as presented in Toriello and Vielma (2012).

4. Fitting Continuous Functions

We present the first convex formulations for fitting PWL functions to discrete data in Sect. 3, for a fixed number of breakpoints B and bounds on the slope and intercept of the PWL function. This mathematical programming formulations utilize only linear functions and can compute optimal PWL functions, for a given set of (discrete) data points. We use the linear formulation $\mathcal{F}^{\max}_{\text{MILP}}(B)$ in this section to compute optimal PWL functions for univariate continuous functions as well. This requires a dynamic and adaptive discretization of the continuous function domain $[\underline{X}, \overline{X}]$ to yield a series of discrete function data. The resulting discrete fitting problems can then be solved using the $\mathcal{F}^{\max}_{\text{MILP}}(B)$ formulation.

Let there be given a univariate continuous function $f(x) : [\underline{X}, \overline{X}] \to \mathbb{R}$ over the compact and non-empty domain $[\underline{X}, \overline{X}]$. For a fixed number of breakpoints B, the task is then to compute a continuous PWL function $p(x) : [\underline{X}, \overline{X}] \to \mathbb{R}$ which optimizes some distance function (or metrics) between $f(\cdot)$ and $p(\cdot)$. We limit the discussion in this paper to the maximum-norm (or maximum difference). Alternatively, the area between $f(x)$ and the PWL function $p(\cdot)$ can be minimized (Kallrath and Rebennack 2014); this metric is the continuum analogue to minimizing the sum of the absolute differences for a set of discrete data points.

For given $f(\cdot)$ and $p(\cdot)$, the maximum difference E can be computed by solving the following continuous optimization problem

$$E := \max_{x\in[\underline{X},\overline{X}]} |p(x) - f(x)| = \max_{b\in\{1,\dots,B-1\}} \max_{x_s\in[X_b,X_{b+1}]} |p(x_s) - f(x_s)|. \tag{20}$$

Note that (20) decomposes into $B-1$ box-constrained, global and univariate optimization problems.

The computation of a PWL function $p(\cdot)$ then requires the minimization of the maximum difference, *i.e.*, the following minimax optimization problem has to be solved

$$\begin{aligned} \eta^* := \min \;\; & \eta \\ \text{s.t.} \;\; & |p(x) - f(x)| \leq \eta \qquad \forall x \in [\underline{X}, \overline{X}] \\ & p(\cdot) \text{ is continuous PWL function on } [\underline{X}, \overline{X}] \text{ with} \\ & \quad \text{slope} \in [\underline{C}, \overline{C}] \text{ and intercept} \in [\underline{D}, \overline{D}]. \end{aligned} \tag{21}$$

Rebennack and Kallrath (2015) proposed two discretizations to resolve the infinite constraint (21). In the first discretization, $\tilde{I}_s$ data points are placed equidistantly in between any two consecutive breakpoints for segment s. Because the breakpoints are decision variables, these data points are also decision variables tied to the associated segment s. Discretized constraints (21) are then potentially non-linear and non-convex. The resulting model, computing an optimal PWL function minimizing the maximum difference, is a non-convex (continuous) non-linear programming (NLP) problem.

In contrast to the first method where the data points depend on the chosen breakpoints, the second method uses an a-priori discretization, *i.e.*, the data points are fixed. Therefore, I data points are introduced, as discretization of domain $[\underline{X}, \overline{X}]$. Then, binary decision variables assign these data points to the appropriate segment. This leads to the case of I discrete data points and the non-convex MIQCP model proposed by Goldberg et al. (2014).

Thus, by applying the a-priori discretization idea, the $\mathcal{F}^{\max}_{\text{MILP}}(B)$ formulation proposed in this paper can be utilized to fit PWL functions to a continuous function $f(\cdot)$. Before we describe the resulting algorithm, we formalize

Definition 2. We call a finite set $\mathcal{I} = \{X_i \mid i = 1, \ldots, I\}$ a *discretization* of the non-empty compact interval $[\underline{X}, \overline{X}] \subset \mathbb{R}$ if

$$\underline{X} = X_1 < \ldots < X_i < X_{i+1} < \ldots < X_I = \overline{X}.$$

Discretization $\mathcal{J}$ of $[\underline{X}, \overline{X}]$ is a *refinement* of discretization $\mathcal{I}$ of $[\underline{X}, \overline{X}]$, if $\mathcal{I} \subset \mathcal{J}$. □

Now, let there be given some discretization $\mathcal{I}$ of $[\underline{X}, \overline{X}]$ together with an optimal PWL function $p^{\mathcal{I}}(x)$ minimizing the (discrete) maximum difference, *e.g.*, computed via the MILP problem $\mathcal{F}^{\max}_{\text{MILP}}(B)$ of Sect. 3. Let the (discrete) maximum difference, for the discrete data points, be denoted by $e^{\text{B}}(\mathcal{I})$, *i.e.*, the optimal objective function value of $\mathcal{F}^{\max}_{\text{MILP}}(B)$. The maximum difference $E^{\text{B}}(\mathcal{I}) \equiv E$ between $f(x)$ and $p^{\mathcal{I}}(x)$ computed over the original problem, as defined in (20), is likely to be greater than $e^{\text{B}}(\mathcal{I})$ but never smaller, *i.e.*,

$$e^{\text{B}}(\mathcal{I}) \leq E^{\text{B}}(\mathcal{I}) \qquad \forall \text{ finite discretizations } \mathcal{I} \text{ of } [\underline{X}, \overline{X}].$$

If, for some chosen tolerance $\varepsilon > 0$,

$$E^{\text{B}}(\mathcal{I}) - e^{\text{B}}(\mathcal{I}) \leq \varepsilon, \tag{22}$$

then $p^{\mathcal{I}}(x)$ is an ε-optimal PWL function for $f(x)$ using B breakpoints minimizing the maximum difference. Otherwise, the discretization $\mathcal{I}$ needs to be refined in a suitable manner. We summarize the resulting method in Algorithm 1.

Algorithm 1 computing PWL function for $f(x)$:

Input: Function $f(x)$, non-empty domain $[\underline{X}, \overline{X}]$, tolerance $\varepsilon > 0$, number of breakpoints $B > 2$, bounds on slope $[\underline{C}, \overline{C}]$ and intercept $[\underline{D}, \overline{D}]$.

Output: Continuous ε-optimal PWL function $p(x)$ with maximum difference $\leq E^{\text{B}} + \frac{\varepsilon}{2}$, slope $\in [\underline{C}, \overline{C}]$ and intercept $\in [\underline{D}, \overline{D}]$.

1. Initialize: Choose discretization $\mathcal{I}$ with $I > B$.

2. Solve MILP: Compute $\frac{\varepsilon}{2}$-optimal continuous PWL function $p^{\mathcal{I}}(x)$ for $\mathcal{I}$ with (discrete) maximum difference $e^{\mathrm{B}}(\mathcal{I})$, slope $\in [\underline{C}, \overline{C}]$ and intercept $\in [\underline{D}, \overline{D}]$. Such a PWL function can be computed with $\mathcal{F}^{\max}_{\mathrm{MILP}}(B)$.

3. Evaluate: Solve global optimization problem (20) to $\frac{\varepsilon}{2}$-optimality to obtain $E^{\mathrm{B}}(\mathcal{I}) \equiv E$.

4. Check optimality: If $E^{\mathrm{B}}(\mathcal{I}) - e^{\mathrm{B}}(\mathcal{I}) \leq \frac{\varepsilon}{2}$ then $p(x) \equiv p^{\mathcal{I}}(x)$, $E^{\mathrm{B}} \equiv E^{\mathrm{B}}(\mathcal{I})$ and STOP.

5. Refinement: Refine discretization $\mathcal{I}$ and GOTO 2.

As initial discretization in step 1, the equidistant placement of the discrete data points is a natural choice, *i.e.*,

$$\mathcal{I} = \left\{ X_i := \underline{X} + (i-1) \cdot \frac{\overline{X} - \underline{X}}{I - 1} \mid i = 1, \ldots, I \right\}.$$

For I, any number $I > B$ suffices. However, to enhance computational speed, I should be chosen such that the resulting discrete fitting problem does not allow for a perfect PWL function, *i.e.*, a PWL function with a (discrete) maximum difference of zero (otherwise, this computational effort is wasted). This is problem dependent as the number of required points I depend on the shape of the function to be approximated.

In step 3, the evaluation of the computed PWL function requires the solution of a global optimization problem (20). In some cases, this problem can be solved analytically; recall that the problems are one-dimensional. In all other cases, a global solver needs to be employed. Thus

Remark 1. If the global optimization problems (20) can be solved analytically, then Algorithm 1 uses only MILP techniques. Otherwise, Algorithm 1 requires a global solver for the class of box-constrained univariate continuous optimization problems. □

The refinement in step 5 crucially affects the computational performance of Algorithm 1 (and its finite convergence, see below). On the one hand, by adding "many" data points to $\mathcal{I}$ we expect a decrease in the number of iterations in Algorithm 1 compared to adding only a "few" data points. On the other hand, the number of (discrete and continuous) decision variables of the resulting MILP, to be solved in step 2, grows linearly with I. Thus, we expect longer solve times for the MILPs with bigger sets $\mathcal{I}$. Consequently, we are facing a tradeoff between the number of iterations in Algorithm 1 and the computational difficulty in solving the resulting MILPs. The refinement strategy we use in our computations utilizes the solutions of the global optimization problems (20). Specifically, for $b=1,\dots,B-1$ we solve

$$E_b^+ := \max_{x_s\in[X_b,X_{b+1}]} p(x_s)-f(x_s) \qquad \text{and} \tag{23a}$$

$$E_b^- := \min_{x_s\in[X_b,X_{b+1}]} p(x_s)-f(x_s). \tag{23b}$$

Let x_b^+ and x_b^- be an optimal solution of (23a) and (23b), respectively. We add x_b^+ to $\mathcal{I}$, if

$$E_b^+ \geq e^{\mathrm{B}}(\mathcal{I}) + \frac{\varepsilon}{3}, \tag{24}$$

where $e^{\mathrm{B}}(\mathcal{I})$ is the discrete maximum difference of $p^{\mathcal{I}}(x)$. Similarly, for x_b^-.

To enhance the computational speed of Algorithm 1, we make two observations. First, because of the Definition 2 of the refinement $\mathcal{J}$ of a discretization $\mathcal{I}$, $e^{\mathrm{B}}(\mathcal{I})$ is a lower bound on the maximum difference of the MILP in the next iteration, *i.e.*,

$$e^{\mathrm{B}}(\mathcal{I}) \leq e^{\mathrm{B}}(\mathcal{J}).$$

This lower bound may aid in solving the MILPs, especially because the lower bound is quite tight for later iterations. Second, feasible solutions of the MILPs can be constructed from solutions of previous iterations. Specifically, the best computed PWL function throughout the algorithm can be used to warm-start the MILP solvers.

Next, we establish the finiteness of Algorithm 1.

Theorem 3. *Let $f(x):[\underline{X},\overline{X}]\to\mathbb{R}$ be a continuous function on $[\underline{X},\overline{X}]\neq\emptyset$, $B>2$, bounds on slope $[\underline{C},\overline{C}]$ and intercept $[\underline{D},\overline{D}]$. Then Algorithm 1 computes a continuous ε-optimal PWL function $p(x):[\underline{X},\overline{X}]\to\mathbb{R}$, with slope $\in[\underline{C},\overline{C}]$ and intercept $\in[\underline{D},\overline{D}]$, in a finite number of iterations if the refinement strategy admits an arbitrarily small distance between consecutive data points.* □

Proof First, per requirement, the refinement strategy admits discretizations $\mathcal{I}$ with arbitrarily small distance between consecutive data points, *i.e.*, for every $\tilde{h}>0$, there exists an $\widetilde{\mathcal{I}}$ with $\max_{i=1,\dots,\tilde{I}-1}\{X_{i+1}-X_i\} \le \widetilde{h}$. Such a discretization $\widetilde{\mathcal{I}}$ will always be constructed through Algorithm 1, if it does not converge before.

Second, for fixed $\mathcal{I}$, the difference between $f(\cdot)$ and $p^{\mathcal{I}}(\cdot)$ in the interval $[X_i, X_{i+1}]$ is bounded by

$$|f(x)-p^{\mathcal{I}}(x)| \le e^{\mathrm{B}}(\mathcal{I}) + \overline{C}\cdot(X_{i+1}-X_i) + \sup\big\{|f(u)-f(v)| \,:\, X_i \le u, v \le X_{i+1}\big\}.$$

The right-hand-side term $e^{\mathrm{B}}(\mathcal{I})$ is given by step 2, $\overline{C}\cdot(X_{i+1}-X_i)$ is the extreme value which any PWL function can take and the right most term is a modulus of continuity. Because $f(\cdot)$ is continuous over the compact interval $[\underline{X},\overline{X}]$, it is also uniformly continuous on $[\underline{X},\overline{X}]$ according to the Heine-Cantor Theorem.

First and second implies that for every $\varepsilon > 0$, there exists an $\hat{h} > 0$ and a finite discretization $\widehat{\mathcal{I}}$, such that

$$\overline{C}\cdot(X_{i+1}-X_i) + \sup\big\{|f(u)-f(v)| \,:\, X_i \le u, v \le X_{i+1}\big\} \le \frac{\varepsilon}{2} \qquad \forall i = 1,\dots,\hat{I}-1.$$

For $\widehat{\mathcal{I}}$ we obtain

$$|f(x)-p^{\widehat{\mathcal{I}}}(x)| \le e^{\mathrm{B}}(\widehat{\mathcal{I}}) + \frac{\varepsilon}{2}. \tag{25}$$

Let $E^{\mathrm{B},*}$ be the minimax among all continuous PWL functions with B breakpoints on $[\underline{X},\overline{X}]$ with slope $\in[\underline{C},\overline{C}]$ and intercept $\in[\underline{D},\overline{D}]$. By construction of the discretized fitting problem in step 2 and because $p^{\widehat{\mathcal{I}}}(x)$ is a feasible PWL function, we obtain together with (25),

$$e^{\mathrm{B}}(\widehat{\mathcal{I}}) - \frac{\varepsilon}{2} \le E^{\mathrm{B},*} \le e^{\mathrm{B}}(\widehat{\mathcal{I}}) + \frac{\varepsilon}{2}$$

Thus

$$|E^{\mathrm{B},*} - e^{\mathrm{B}}(\widehat{\mathcal{I}})| \le \frac{\varepsilon}{2}.$$

Because of step 3, we have

$$|E^{\mathrm{B}}(\widehat{\mathcal{I}}) - e^{\mathrm{B}}(\widehat{\mathcal{I}})| \le \frac{\varepsilon}{2}$$

and finally

$$|E^{\mathrm{B},*} - E^{\mathrm{B}}(\widehat{\mathcal{I}})| \le \varepsilon.$$

Thus, $p^{\widehat{\mathcal{I}}}(x)$ is an ε-optimal continuous PWL function which is computed after finitely many iterations. □

Note that the discussed refinement strategy above via (24) does not allow arbitrarily small distances between consecutive data points. However, it does allow arbitrarily small distances between data points where computed PWL functions violate the fitting criteria. As such, the presented refinement strategy is valid and leads to a finitely convergent algorithm for fitting continuous PWL functions to continuous univariate functions.

As a final remark, the choice of $[\underline{C},\bar{C}]$ and $[\underline{D},\bar{D}]$ impacts the approximation quality of the PWL function computed, just like the number of breakpoints B and the tolerance $\varepsilon > 0$ does. This is the case, because Algorithm 1 computes an optimal PWL function, respecting the bounds on the slope, $[\underline{C},\bar{C}]$, and the intercept, $[\underline{D},\bar{D}]$.

5. Computational Results

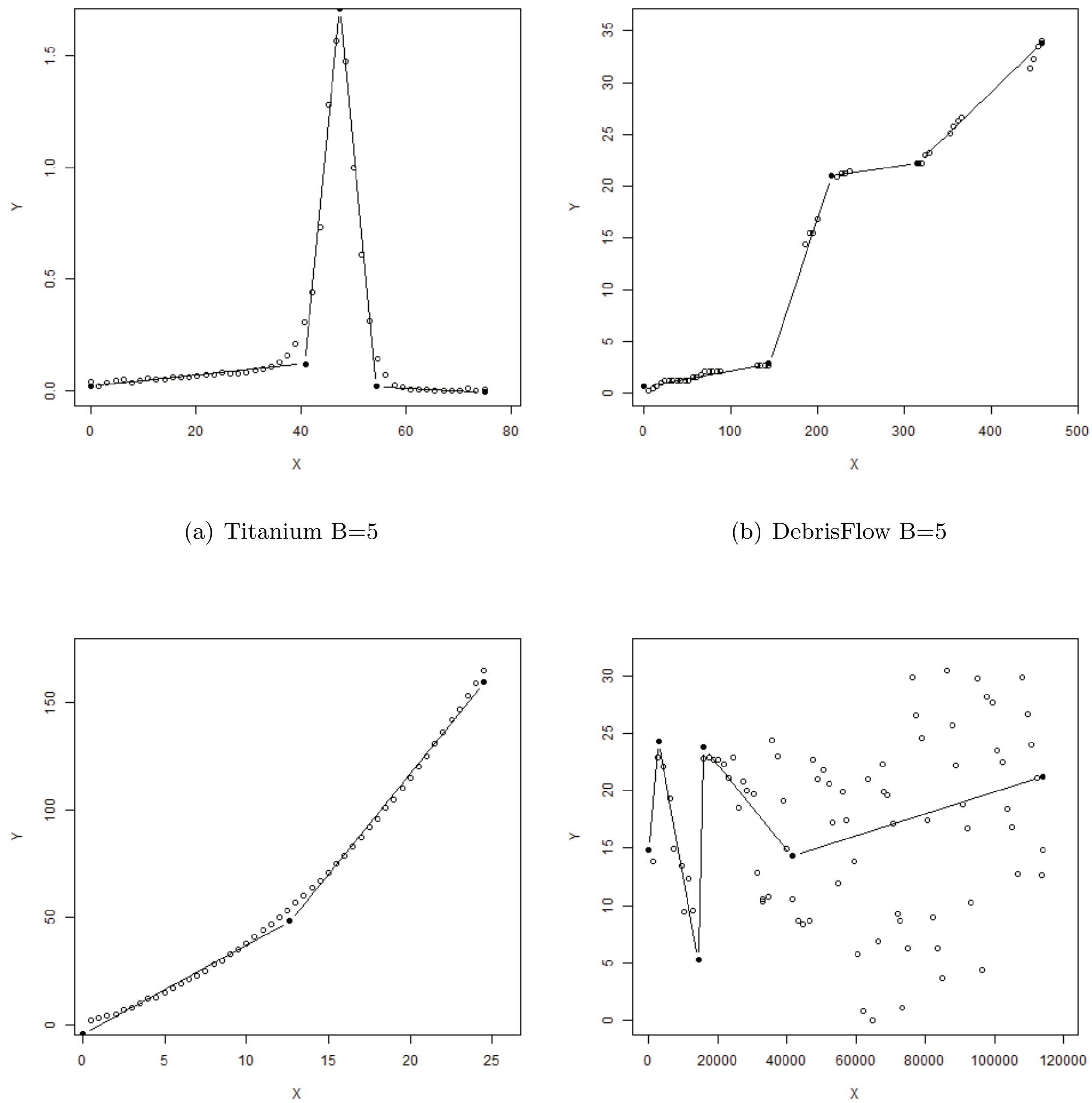
In this section we present computational results for fitting discrete data and univariate continuous functions.

5.1. Discrete Data

We run the models with GAMS 24.7.1 on an Intel 3.5 GHz machine with 32 GB of RAM. Stopping criteria of an absolute optimality gap of 0.001 is used.

We benchmark our convex formulations on six different data sets with a total of 117 instances. The "Titanium" data ($I = 49$) is a classical data set often used in spline approximation research (De Boor and Rice 1968, Jupp 1978). The data represents a thermal property of titanium but has been scaled (Jupp 1978). The "DebrisFlow" data ($I = 44$) is taken from a case study of post-fire debris flow hazard management around the Santa Barbara area following a 2009 wildfire (McCoy et al. 2016). The x-coordinates in the data set represent debris flow volume in thousands of cubic meters while the y-coordinates represent expected structural damages in dollars for a particular drainage basin. The "Mpstorage50" data set ($I = 50$) represents a subset of a data set with the x-values equal to water elevation at the Morrow Point reservoir and y-values equal to the volume of the reservoir. This data set is used in Goldberg et al. (2014). The raw material height data set "Rmheight" ($I = 84$) represents the height of plastic pellets in a tall narrow container, measured over a period of three months with the measurement time converted to integers[2]. The "Paperweight" data

[2] This data set is available from OpenMV.net.



(a) Titanium B=5

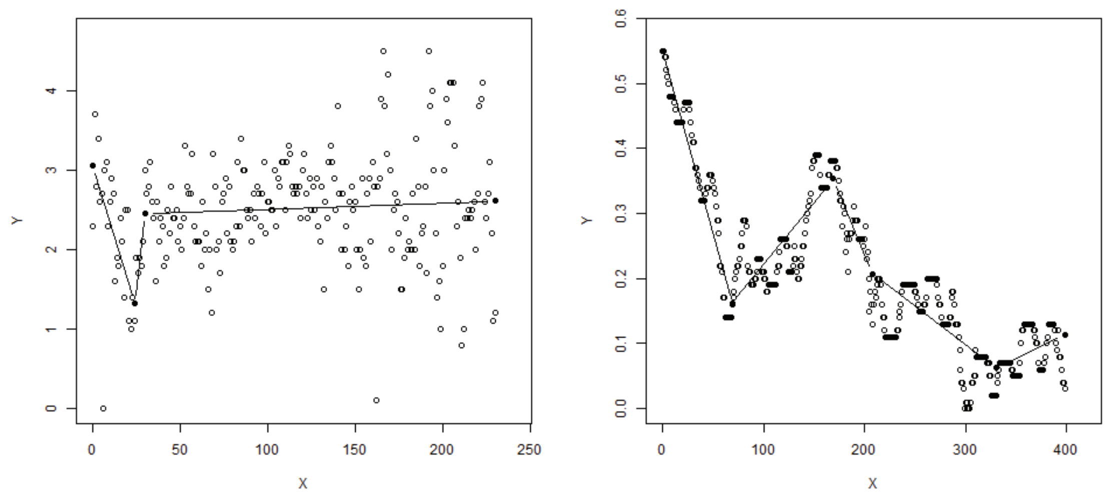
(b) DebrisFlow B=5

(c) Mpstorage50 B=3

(d) Rmheight B=6

Figure 2 Continuous PWL functions for the six different data sets minimizing the sum of absolute differences.

set ($I = 231$) represents a measure of paper density taken from an online scanning gauge 30 seconds apart at a large paper manufacturer (MacGregor and Harris 1993). Finally, the



(e) Paperweight B=4 (f) Elevationbig B=6

Figure 2 continued.

"Elevationbig" data set ($I = 1216$) provides a large number of data points (Goldberg et al. 2014). Fig. 2(a)-(f) illustrate the data sets together with some computed PWL function.

Tables 3-5 present the computational results for the six different data sets, different number of breakpoints B, different solvers for the convex and non-convex models and the three different distance metrics. The solve times are reported in seconds. If the solver does not terminate within the time limit, then the lower and upper bounds are reported in the table for the objective function value. The best results for each category is marked in bold face. The procedure of iteratively adding constraints (15c)-(15h) and solving the resulting convex models with CPLEX is denoted by CPLEX-P.

Table 3 Sum of Absolute Differences: Computational results. * Time limit of 3,600 sec. reached.

				B								
			Solver	3	4	5	6	7	8	9	10	11
Titanium	Objective	$\mathcal{F}^{\text{abs}}_{\text{MILP}}(B)$	CPLEX	7.26	5.74	1.08	0.74	0.49	0.37	0.27	0.18	(0.09, 0.16)
			CPLEX-P	7.26	5.74	1.08	0.74	0.49	0.37	0.27	0.18	0.15
			GUROBI	7.26	5.74	1.08	0.74	0.49	0.37	0.27	0.18	0.15
			MOSEK	7.26	5.74	1.08	0.74	0.49	0.37	0.27	0.18	(0.07, 0.15)
		$\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$	BARON	7.26	5.74	1.08	0.74	0.49	0.37	0.27	0.18	(0.13, 0.15)
			GLOMIQO	7.26	5.74	1.08	0.74	0.49	0.37	0.27	0.18	(0.13, 0.18)
			LINDOGLOBAL	7.26	5.74	1.08	0.74	0.49	(0.14, 0.46)	(0.02, 0.39)	(0, 0.34)	(0, 0.63)
			SCIP	7.26	5.74	1.08	0.74	0.49	0.37	0.27	0.18	0.15
	Solve time	$\mathcal{F}^{\text{abs}}_{\text{MILP}}(B)$	CPLEX	**0.1**	**0.4**	0.5	3.7	8.1	182.3	275.4	475.2	*
			CPLEX-P	0.8	1.1	0.5	1.2	2.9	**16.0**	45.1	329.0	604.1
			GUROBI	0.2	0.6	0.9	1.8	7.4	51.0	149.8	298.4	489.3
			MOSEK	0.6	4.2	4.1	93.4	114.0	531.0	2,464.3	2,239.3	*
		$\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$	BARON	1.2	6.2	0.6	3.6	3.4	215.1	310	704.2	*
			GLOMIQO	0.9	7.8	**0.4**	**0.8**	**4.0**	245.1	989.3	2,562.5	*
			LINDOGLOBAL	8.3	191.3	136	332.5	947.2	*	*	*	*
			SCIP	0.7	3.3	3.3	3.6	9.0	19.4	**37.6**	**51.4**	**187.1**
DebrisFlow	Objective	$\mathcal{F}^{\text{abs}}_{\text{MILP}}(B)$	CPLEX	66.66	27.15	10.75	8.85	7.21	6.2	4.55	(2.73, 3.9)	2.71
			CPLEX-P	66.66	27.15	10.75	8.85	7.21	6.2	4.55	3.9	2.71
			GUROBI	66.66	27.15	10.75	8.85	7.21	6.2	4.55	3.9	2.71
			MOSEK	66.66	27.15	10.75	8.85	7.21	6.2	(1, 4.55)	(0.64, 3.92)	(0, 3.45)
		$\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$	BARON	66.66	27.15	10.75	8.85	7.21	6.2	4.55	(2.59, 4.41)	(1.77, 2.73)
			GLOMIQO	66.66	27.15	10.75	8.85	7.21	6.2	4.55	(3.41, 4.2)	(2.38, 2.71)
			LINDOGLOBAL	66.66	27.15	10.75	8.85	(6.72, 7.21)	(1.18, 6.2)	(0, 5.38)	(0, 3.91)	(0, 4.95)
			SCIP	66.66	27.15	10.75	8.85	7.21	6.2	4.55	3.9	(2.6, 2.71)
	Solve time	$\mathcal{F}^{\text{abs}}_{\text{MILP}}(B)$	CPLEX	**0.1**	**0.2**	**0.4**	6.9	16.5	**86.8**	1,437.8	*	3,176.7
			CPLEX-P	0.5	0.7	0.8	3.6	15.0	102.8	251.8	2,611.8	1,093.3
			GUROBI	**0.1**	0.4	0.5	**2.4**	**12.9**	53.1	**60.7**	1,188.9	**802.2**
			MOSEK	0.5	1.5	4.4	29.9	435.0	2,091.7	*	*	*
		$\mathcal{F}^{\text{abs}}_{\text{MIQCP}}(B)$	BARON	1.0	1.8	5.5	60.6	239.0	2,420.0	3,396.9	*	*
			GLOMIQO	0.6	2.7	5.4	32.3	140.3	1,695.0	1,995.7	*	*
			LINDOGLOBAL	22.2	151.5	421.5	2,460.0	*	*	*	*	*
			SCIP	0.5	1.0	2.3	7.4	31.8	160.2	166.5	**959.1**	*

Table 3 continued. † Time limit of 86,400 sec. reached.

				B								
			Solver	3	4	5	6	7	8	9	10	11
Mpstorage50	Objective	$\mathcal{F}^{abs}_{MILP}(B)$	CPLEX	100.08	43.18	24.51	16.48	12.34	10.4	9	7.8	(5.37, 7)
			CPLEX-P	100.08	43.18	24.51	16.48	12.34	10.4	9	7.8	7
			GUROBI	100.08	43.18	24.51	16.48	12.34	10.4	9	7.8	7
		$\mathcal{F}^{abs}_{MIQCP}(B)$	SCIP	100.08	43.18	24.51	16.48	12.34	10.4	9	7.8	(6.37, 7)
	Solve time	$\mathcal{F}^{abs}_{MILP}(B)$	CPLEX	**0.1**	**0.3**	0.9	5.6	21.5	149.4	549.7	9,392.8	†
			CPLEX-P	0.6	**0.3**	**0.8**	4.8	15.3	**64.4**	**264.2**	**921.2**	11,081.5
			GUROBI	**0.1**	0.4	0.9	**3.7**	**14.7**	100.5	509.9	4,205.2	**3,872.7**
		$\mathcal{F}^{abs}_{MIQCP}(B)$	SCIP	0.4	1.0	2.8	5.2	22.5	106.5	965.4	6,834.5	†
Rmheight	Objective	$\mathcal{F}^{abs}_{MILP}(B)$	CPLEX	508.2	481.13	450.81	433.51	401.39				
			CPLEX-P	508.2	481.13	450.81	433.49	401.39				
			GUROBI	508.2	481.13	450.81	433.49	401.39				
		$\mathcal{F}^{abs}_{MIQCP}(B)$	SCIP	508.2	481.13	450.82	(408.44, 441.24)	(394.38, 411.08)				
	Solve time	$\mathcal{F}^{abs}_{MILP}(B)$	CPLEX	**0.2**	**4.0**	49.1	957.7	9,118.7				
			CPLEX-P	1.8	12.3	193.5	693.4	**3,920.0**				
			GUROBI	0.3	5.3	**31.3**	**531.8**	7,632.3				
		$\mathcal{F}^{abs}_{MIQCP}(B)$	SCIP	2.8	788.3	2,529.9	†	†				
Paperweight	Objective	$\mathcal{F}^{abs}_{MILP}(B)$	CPLEX	116.88	113.55	110.77	(106.27, 107.76)					
			CPLEX-P	116.88	113.55	110.77	(102.81, 107.76)					
			GUROBI	116.86	113.55	110.77	(104.96, 107.76)					
		$\mathcal{F}^{abs}_{MIQCP}(B)$	SCIP	116.88	113.56	110.77	(103.15, 108.44)					
	Solve time	$\mathcal{F}^{abs}_{MILP}(B)$	CPLEX	**0.7**	**30.1**	1,943.0	†					
			CPLEX-P	8.1	185.4	**1,003.5**	†					
			GUROBI	2.0	32.2	1,174.9	†					
		$\mathcal{F}^{abs}_{MIQCP}(B)$	SCIP	9.3	92.5	7,865.3	†					
Elevationbig	Objective	$\mathcal{F}^{abs}_{MILP}(B)$	CPLEX	19.77	16.48	13.71	12.43					
			CPLEX-P	19.77	16.48	13.71	12.43					
			GUROBI	19.77	16.48	13.71	(10.68, 12.43)					
		$\mathcal{F}^{abs}_{MIQCP}(B)$	SCIP	19.77	16.48	13.71	(11.53, 12.81)					
	Solve time	$\mathcal{F}^{abs}_{MILP}(B)$	CPLEX	**1.6**	59.9	2,073.0	66,267.0					
			CPLEX-P	20.5	747.3	1,515.8	**25,157.9**					
			GUROBI	1.9	**50.6**	**1,259.8**	†					
		$\mathcal{F}^{abs}_{MIQCP}(B)$	SCIP	10.5	355.8	23,269.4	†					

Table 4 Sum of Squared Differences: Computational results. * Time limit of 3,600 sec. reached.

				B						
			Solver	3	4	5	6	7	8	9
Titanium	Objective	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	CPLEX	3.78	2.13	0.07	0.03	0.02	0.01	0
			CPLEX-P	3.78	2.13	(0, 3.61)	0.03	0.02	0.01	0
			GUROBI	3.78	2.13	0.07	0.04	0.02	0.01	0
			MOSEK	3.78	2.13	0.07	0.03	0.02	0.01	(0, 0.06)
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	BARON	3.78	2.13	0.07	0.03	0.02	0.01	0
			GLOMIQO	(2.31, 3.78)	(0.73, 2.13)	0.07	0.03	0.02	0.01	0
			LINDOGLOBAL	3.78	(2.1, 2.13)	0.07	0.03	0.02	0.01	(0, 0.02)
			SCIP	3.78	2.13	0.07	0.04	(0, 0.03)	(0, 0.03)	0
	Solve time	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	CPLEX	**0.1**	0.8	2.5	2.1	13.9	30.2	90.9
			CPLEX-P	26.8	41.5	*	143	125.6	648.3	*
			GUROBI	0.2	**0.4**	**0.9**	**0.8**	**3.2**	**8.0**	**30.2**
			MOSEK	0.9	14.1	38.3	193.9	1,124.8	1,093.0	*
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	BARON	3.5	7.4	3.6	8.8	21.0	36.0	194.1
			GLOMIQO	*	*	6.1	686.3	*	1,116.1	2,896.4
			LINDOGLOBAL	232.9	*	85.2	773.7	1,514.4	3,219.8	*
			SCIP	1.3	9.3	5.6	1,089.6	*	*	120.4
DebrisFlow	Objective	$\mathcal{F}^{\text{max}}_{\text{MILP}}(B)$	CPLEX	290.51	37.96	5.06	3.9	2.87	1.85	1.04
			CPLEX-P	290.51	37.96	5.06	3.9	(0, 37.77)	1.85	1.04
			GUROBI	290.51	37.96	5.06	3.9	2.87	1.85	1.04
			MOSEK	290.51	37.96	5.06	3.9	2.87	1.85	(0.01, 1.91)
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	BARON	290.51	37.96	5.06	3.9	2.87	1.85	(0.8, 1.04)
			GLOMIQO	(217.97, 290.57)	(4.13, 37.96)	(2.93, 5.06)	(2.11, 4.01)	(1.39, 2.89)	(0.57, 1.97)	(0.36, 1.34)
			LINDOGLOBAL	(208.97, 290.51)	(37.89, 37.96)	(4.84, 5.06)	(2.91, 3.9)	(1.71, 2.87)	(0.76, 1.85)	(0.07, 1.94)
			SCIP	290.51	37.96	5.06	3.9	2.87	1.85	(0, 2.91)
	Solve time	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	CPLEX	0.2	**0.4**	0.8	5.4	16.8	71.7	**63.8**
			CPLEX-P	224.5	20.0	343.7	333.8	*	783.3	2,265.9
			GUROBI	**0.1**	**0.4**	**0.5**	**3.6**	**10.0**	**50.5**	164.4
			MOSEK	0.9	3.7	41.4	212.0	1,418.2	2,742.2	*
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	BARON	2.7	4.3	7.6	32.6	204.7	1,993.1	*
			GLOMIQO	*	*	*	*	*	*	*
			LINDOGLOBAL	*	*	*	*	*	*	*
			SCIP	1.4	3.4	34.2	559.4	610.9	*	*

Table 4 continued. † Time limit of 86,400 sec. reached.

			Solver	B 3	4	5	6	7
Mpstorage50	Objective	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	CPLEX	289.88	55.79	18.61	8.41	5.46
			CPLEX-P	289.88	55.79	18.61	8.41	(0, 5.6)
			GUROBI	289.88	55.79	18.61	8.41	5.46
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	SCIP	289.88	55.79	18.61	8.41	5.46
	Solve time	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	CPLEX	0.1	0.4	1.1	3.4	17.3
			CPLEX-P	5.4	302.0	22,349.2	52,642.2	†
			GUROBI	**0.2**	**0.3**	**0.6**	**2.1**	**6.6**
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	SCIP	0.6	1.7	5.9	15.1	95.3
Rmheight	Objective	$\mathcal{F}^{\text{max}}_{\text{MILP}}(B)$	CPLEX	4,208.65	3,971.18	3,713.29	3,216.47	
			CPLEX-P	4,208.65	3,971.18	3,713.29	3,216.47	
			GUROBI	4,271.58	(1,530.35, 4,107.26)	(12.80, 4011.43)		
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	SCIP	4,208.65	(3,904.70, 3,971.38)	3,713.29		
	Solve time	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	CPLEX	**0.7**	**15.3**	**118.3**	**1,114**	
			CPLEX-P	73.3	1,438.9	2,411.9	7,039.7	
			GUROBI	27.3	1,218.8	†	†	
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	SCIP	102.3	†	38,097.7	†	
Paperweight	Objective	$\mathcal{F}^{\text{max}}_{\text{MILP}}(B)$	CPLEX	105.62	102.62	97		
			CPLEX-P	(0, 110.63)	102.62	(0, 108.89)		
			GUROBI	105.62	102.62	97		
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	SCIP	105.62	(0, 103.3)	(0, 104.42)		
	Solve time	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	CPLEX	**1.0**	119.3	1,810.3		
			CPLEX-P	†	1,385.2	†		
			GUROBI	**1.0**	**54.1**	**849.0**		
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	SCIP	12.9	†	†		
Elevationbig	Objective	$\mathcal{F}^{\text{max}}_{\text{MILP}}(B)$	CPLEX	1.66	1.08	0.74		
			CPLEX-P	(0, 2.17)	(0, 2.2)	(0, 2.06)		
			GUROBI	1.66	1.08	0.74		
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	SCIP	1.66	1.08	(0, 1.33)		
	Solve time	$\mathcal{F}^{\text{sqr}}_{\text{MIQP}}(B)$	CPLEX	3.8	226.2	†		
			CPLEX-P	†	†	†		
			GUROBI	**3.2**	**126.1**	**1,529.5**		
		$\mathcal{F}^{\text{sqr}}_{\text{MIQCQP}}(B)$	SCIP	49.4	2,734.3	†		

Table 5 Maximum Difference: Computational results. * Time limit of 3,600 sec. reached.

				B								
			Solver	3	4	5	6	7	8	9	10	11
Titanium	Objective	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	0.55	0.49	0.08	0.06	0.05	0.02	0.02	0.01	0.01
			CPLEX-P	0.55	0.49	0.08	0.06	0.05	0.02	0.02	0.01	0.01
			GUROBI	0.55	0.49	0.08	0.06	0.05	0.02	0.02	0.01	0.01
			MOSEK	0.55	0.49	0.08	0.06	0.05	0.02	0.02	0.01	0.01
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	BARON	0.55	0.49	0.08	0.06	0.05	0.02	0.02	0.01	0.01
			GLOMIQO	0.55	0.49	0.08	0.06	0.05	0.02	0.02	0.01	0.01
			LINDOGLOBAL	0.55	0.49	0.08	0.06	0.05	0.02	0.02	(0.01, 0.02)	(0, 0.05)
			SCIP	0.55	0.49	0.08	0.06	0.05	0.02	0.02	0.01	0.01
	Solve time	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	**0.1**	**0.2**	0.5	**0.6**	**1.1**	2.1	6.6	**3.1**	6.7
			CPLEX-P	0.3	0.7	0.4	1.0	1.6	2.7	**2.6**	3.7	7.8
			GUROBI	0.2	0.4	0.6	1.1	1.2	**1.8**	4.3	5.7	**3.0**
			MOSEK	0.2	2.1	5.6	10.3	78.0	213.4	173.4	2689.5	776.9
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	BARON	0.2	1.4	2.3	1.6	2.9	8.1	18	65.4	36.6
			GLOMIQO	0.3	6.3	**0.3**	7.1	12.5	3.8	89.9	15.3	145.9
			LINDOGLOBAL	0.4	21.1	25.5	101.4	1,085.3	376.2	2,075.8	*	*
			SCIP	0.3	0.6	1.5	1.8	3.2	5.6	7.6	14.7	36.2
DebrisFlow	Objective	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	5.38	2.04	0.78	0.65	0.5	0.42	0.39	0.3	0.26
			CPLEX-P	5.38	2.04	0.78	0.65	0.5	0.42	0.39	0.3	0.26
			GUROBI	5.38	2.04	0.78	0.65	0.5	0.42	0.39	0.3	0.26
			MOSEK	5.38	2.04	0.78	0.65	0.5	0.42	0.39	0.3	(0, 0.26)
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	BARON	5.38	2.04	0.78	0.65	0.5	0.42	0.39	(0.29, 0.3)	(0.17, 0.27)
			GLOMIQO	5.38	2.04	0.78	0.65	0.5	0.42	(0.35, 0.39)	(0.28, 0.3)	(0.23, 0.26)
			LINDOGLOBAL	5.38	2.04	0.78	0.65	(0.41, 0.5)	(0.3, 0.5)	(0.29, 0.39)	(0.19, 0.3)	(0.16, 0.26)
			SCIP	5.38	2.04	0.78	0.65	0.5	0.42	0.39	0.3	(0.259, 0.264)
	Solve time	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	**0.1**	**0.2**	**0.2**	0.7	1.5	3.5	5.5	17.2	47.8
			CPLEX-P	0.3	0.4	0.5	0.9	1.7	3.4	**4.5**	8.5	26.5
			GUROBI	0.2	0.3	0.3	**0.3**	**0.7**	**1.5**	9.8	**6.6**	**16.2**
			MOSEK	0.4	0.8	1.9	20.9	53.1	329.4	543.6	3,277.0	*
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	BARON	0.2	0.6	1.0	1.5	17.2	232.2	2,242.7	*	*
			GLOMIQO	0.4	0.7	1.3	3.0	2,422.5	1,191.3	*	*	*
			LINDOGLOBAL	0.6	2.3	9.8	86.3	*	*	*	*	*
			SCIP	0.4	0.4	0.6	1.1	9.9	44.6	122.7	238.9	*

Table 5 continued. † Time limit of 86,400 sec. reached.

				B											
			Solver	3	4	5	6	7	8	16	17	18	19	20	21
Mpstorage50	Objective	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	3.94	1.79	1.00	0.67	0.50	0.50	0.25	0.25	0.25	0.25	0.25	0.25
			CPLEX-P	3.94	1.79	1.00	0.67	0.50	0.50	0.25	0.25	0.25	0.25	0.25	0.25
			GUROBI	3.94	1.79	1.00	0.67	0.50	0.50	0.25	0.25	0.25	0.25	0.25	0.25
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	SCIP	3.94	1.79	1.00	0.67	0.50	0.50	0.25	0.25	0.25	0.25	(0.22, 0.25)	(0.1, 0.25)
	Solve time	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	**0.1**	**0.2**	**0.2**	**0.2**	**0.3**	0.9	11.6	11.0	13.9	655.7	501.1	3,162.8
			CPLEX-P	0.2	**0.2**	0.5	0.6	1.4	2.3	55.9	67.8	72.4	78.0	148.8	**180.6**
			GUROBI	**0.1**	**0.2**	**0.2**	**0.2**	0.5	1.0	**6.2**	**11.7**	**12.5**	**27.6**	**54.4**	223.7
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	SCIP	0.2	**0.2**	0.3	**0.2**	0.6	**0.8**	56.9	63.0	210.3	15,061.0	†	†
Rmheight	Objective	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	14.19	13.37	13.37	12.91	11.58	11.41						
			CPLEX-P	14.19	13.37	13.37	12.91	11.58	11.36						
			GUROBI	14.19	13.37	13.37	12.91	11.58	11.41						
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	SCIP	14.19	13.37	(13.18, 13.37)	(11.99, 13.28)	(11.41, 11.58)	(11.24, 11.41)						
	Solve time	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	**0.2**	**0.4**	**5.2**	18.8	20.1	442.0						
			CPLEX-P	0.4	0.9	6.2	**16.4**	**11.1**	25.3						
			GUROBI	**0.2**	0.5	5.7	24.4	17.8	**16.7**						
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	SCIP	0.7	1.7	†	†	†	†						
Paperweight	Objective	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	2.17	1.85	1.85	1.73	1.67	1.65						
			CPLEX-P	2.17	1.85	1.85	1.73	1.67	1.65						
			GUROBI	2.17	1.86	1.85	1.73	1.67	1.64						
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	SCIP	2.17	1.86	1.85	1.73	1.67	(1.64, 1.65)						
	Solve time	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	1.4	**4.5**	100.1	53.8	4,166.0	7,661.0						
			CPLEX-P	**0.8**	11.3	121.0	**40.5**	1,097.8	2,591.8						
			GUROBI	1.6	9.8	**76.1**	44.8	**1,063.9**	**1,563.3**						
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	SCIP	2.1	12.6	94.9	77.0	2,308.2	†						
Elevationbig	Objective	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	0.15	0.12	0.09	0.08	(0.07, 0.08)	0.07						
			CPLEX-P	0.15	0.12	0.09	0.08	(0.07, 0.08)	0.07						
			GUROBI	0.15	0.12	0.09	0.08	0.08	0.07						
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	SCIP	0.15	0.12	0.09	0.08	0.08	0.07						
	Solve time	$\mathcal{F}^{\max}_{\text{MILP}}(B)$	CPLEX	**1.2**	21.6	**30.0**	72.7	†	46,386.5						
			CPLEX-P	4.9	48.9	43.2	1,433.1	†	†						
			GUROBI	2.5	**20.6**	67.0	658.3	13,365.1	20,348.8						
		$\mathcal{F}^{\max}_{\text{MIQCP}}(B)$	SCIP	8.5	23.6	30.8	**52.8**	**3,096.4**	**335.4**						

Table 6 MILP/MIQP vs. MIQC(Q)P approaches: number of times each approach solved fastest.

		Sum of Absolute Differences	**Sum of Squared Differences**	**Maximum Difference**	**Sum**
# instances		40	29	48	**117**
MILP/MIQP	CPLEX	13	8	19	
	CPLEX-P	9	0	7	
	GUROBI	13	23	22	
	Total	**32**	**29**	**44**	**105**
MIQC(Q)P	GLOMIQO	3	0	1	
	SCIP	4	0	6	
	Total	**7**	**0**	**7**	**14**

The first three data sets ("Titanium," "DebrisFlow" and "Mpstorage50") tend to yield easier problems for the PWL function fitting problems. In contrast, data sets "Rmheight," "Paperweight" and "Elevationbig" yield challenging to very challenging fitting problems. Among the three tested MILP solvers both CPLEX and GUROBI tend to do well, while we observe that specific solvers tend to do better on specific data sets. For some of the problem instances with larger number of breakpoints, CPLEX-P performs best. We test the four different global solvers available in GAMS: BARON (Tawarmalani and Sahinidis 2005), GLOMIQO (Misener and Floudas 2013), LINDOGLOBAL (Schrage 2004) and SCIP (Achterberg 2009). ANTIGONE is not tested because GLOMIQO is a tailored version for quadratic problems. Remarkably, SCIP performs consistently well among the data sets, followed by GLOMIQO.

The MILP approaches outperform the non-convex models in most instances. For the sum of squared differences, the best convex MIQP approaches tend to be faster by one order of magnitude compared to the best non-convex models. The performance of the solvers for each of the three different metrics is summarized in Table 6. Ties in computational speed are counted multiple times. Overall, among the 117 instances, the proposed convex approaches solve the fitting problems on 105 instances faster compared to the non-convex models. The non-convex models outperform the convex models only on 14 instances (out of the 117), containing 2 ties. All instances, which can be solved by the non-convex model formulations, can also be solved by the convex models by at least one solver within the specified time limit. In contrast, 17 instances could not be solved with the non-convex models within the time limit.

5.2. Continuous Functions

Next, we present computational results for univariate, continuous functions. We use seven different functions taken from the literature. They are summarized in Table 7 and plotted in Fig. 3. Functions 1, 3 and 4 are not included in this study; analytical solutions are known for function 1 (Rebennack 2016b) while functions 3 and 4 do not pose a computational challenge. The computations in this section are executed on a standard computer using GAMS 24.8. All MILP problems are solved with CPLEX and the global optimization problems are solved with LINDOGLOBAL.

Table 7 **Continuous functions from Rebennack and Kallrath (2015).**

#	$\mathbf{f(x)}$	$\mathbf{X}_-$	$\mathbf{X}_+$
2	$\ln x$	1	32
5	$\frac{\sin(x)}{x}$	1	12
6	$2x^2 + x^3$	-2.5	2.5
7	$e^{-x}\sin(x)$	-4	4
8	$e^{-100(x-2)^2}$	0	3
9	$1.03e^{-100(x-1.2)^2} + e^{-100(x-2)^2}$	0	3
10	Maranas and Floudas (1994)	0	2π

Table 8 lists the minimal number of breakpoints B^* required, to achieve a given maximum difference of δ, *i.e.*, $E^{B^*} \leq \delta < E^{B^*-1}$. The computational budget for each single optimization problem is set to 24 hours. For each function, the four different values 0.1, 0.05, 0.01 and 0.005 of δ are provided. Columns 3-9 in Table 8 report the results of this paper, while columns 10-12 list the results published in Rebennack and Kallrath (2015). We abbreviate their approach with GO in the following discussion. B_- and B_+ are lower and upper bounds on B^*, if B^* cannot be determined. "iter" is the number of iterations of Algorithm 1 until convergence; see below. The total computational times of Algorithm 1 are split up into the time to solve the MILPs and the global optimization problems.

For given maximum difference δ and number of breakpoints B, we can change the stopping criterion of Algorithm 1 to determine that $B+1 \leq \mathrm{B}^*$ or $\mathrm{B}^* \leq B$ as follows. Algorithm 1 either terminates once the lower bound on the maximal difference exceeds δ or a PWL function with maximal difference $\leq \delta$ is computed. This way, Algorithm 1

Table 8 **Minimal number of breakpoints for given accuracy δ. $-$: global solution was obtained analytically; bold-face marks significant improvement; italics are better solutions computed.**

		This Paper: Algorithm 1						Reb&Kal 2015		
#	δ	B^*	B_-	iter	time [sec.] MILP	global	total	B^*	B_-	B_+
2	0.100	4		2	0.1	-	0.1	4		
	0.050	5		3	0.1	-	0.1	5		
	0.010	10		7	9.8	-	9.8	10		
	0.005	14		10	128.2	-	128.3	14		
5	0.100	4		1	0.0	0.3	0.7	4		
	0.050	6		6	0.6	3.0	6.5	6		
	0.010	10		12	27.1	11.3	49.7	10		
	0.005	13		5	22.8	6.4	35.8	13		
6	0.100	12		5	24.3	-	24.4	12		
	0.050	16		7	107.5	-	107.7	16		
	0.010		**32**	5	6,818.8	-	6,819.1		16	35
	0.005		**41**	6	35,731.8	27.1	35,787.4		16	48
7	0.100	**15**		10	279.4	16.7	311.5		5	15
	0.050	**20**		8	14,480.1	17.9	14,514.6		5	20
	0.010		**35**	6	35,361.5	25.2	35,411.0		5	44
	0.005		**36**	3	70,288.7	12.5	70,313.1		5	62
8	0.100	5		2	0.1	0.8	1.7	5		
	0.050	***6***		5	0.5	2.3	5.3		5	7
	0.010	**12**		5	48.2	5.6	59.6		5	12
	0.005	**15**		4	234.9	6.3	247.2		5	15
9	0.100	8		1	0.3	0.8	1.9	8		
	0.050	***10***		4	4.2	3.9	12.0		8	12
	0.010	**22**		8	13,836.1	19.0	13,873.4		8	22
	0.005	***28***		7	42,036.2	22.2	42,068.4		8	29
10	0.100	**17**		9	3,034.8	26.2	3,085.5		4	17
	0.050	***22***		12	110,634.1	51.1	110,735,3		4	23
	0.010		**43**	16	82,609.3	111.8	82,831.3		4	46
	0.005		**51**	7	89,912.4	57.3	90,028.4		4	67

can be used to compute the minimal number of required breakpoints B^* for a maximal difference of δ.

From the results in Table 8 we observe that Algorithm 1 can compute B^* for all instances, where the GO approach can compute them. Out of the 16 instances where B^* cannot be computed by GO, Algorithm 1 can determine B^* for 10 instances!

Table 9 **Lower and upper bound on minimax difference E^{B}, given the number of breakpoints. ‡: value not provided; †: time limit reached; ∗ wrongly reported values; −: global solution was obtained analytically; bold-face marks significant improvement.**

#	B	This Paper: Algorithm 1 E^{B}_{-}	E^{B}_{+}	iter	time [sec.] MILP	global	all	Reb&Kal 2015 E^{B}_{-}	E^{B}_{+}
2	4	0.081872	0.081966	15	1.1	-	1.1	0.081899	0.081922
	5	0.046422	0.046491	18	2.7	-	2.8	0.046281	0.046595
	10	0.009228	0.009291	27	235.0	-	235.4	0.009211	0.009287
	14	0.004412	0.004512	33	1,534.9	-	1,535.9	0.004429	0.004446
5	4	0.051382	0.051400	11	0.6	4.2	8.5	0.051237	0.051847
	6	**0.019835**	**0.019903**	15	2.6	8.6	20.0	0.018513	0.022101
	10	**0.009507**	0.009590	10	24.3	10.6	45.5	0	‡
	13	**0.004180**	0.004276	16	180.8	23.4	227.0	0	‡
6	12	**0.086732**	**0.086832**	37	1,772.3	-	1,774.5	0.085288	0.095080
	16	**0.045264**	0.045349	35	3,788.4	-	3,791.0	0	‡
	35	**0.007076**	0.018208	3	20,626.1	-	20,626.3†	0	‡
7	15	**0.087564**	0.087658	45	7,218.5	79.1	7,373.0	0	‡
	20	**0.032211**	0.150745	3	18,579.0	8.6	18,595.8†	0	‡
8	5	**0.054068**	**0.054152**	9	1.0	4.1	9.4	0.053910	0.054603
	6	0.043749	**0.043841**	19	7.7	11.1	29.8	‡	‡
	7	**0.042315**	**0.042404**	28	27.2	18.8	65.0	0.009178	0.990842
	12	**0.007878**	**0.007968**	15	248.7	18.6	286.4	0.009158*	0.990842
	15	**0.004749**	0.004848	16	830.3	26.0	882.6	0	‡
9	8	**0.055690**	**0.055785**	26	19.9	22.1	63.3	0.085773*	0.941691
	10	0.046137	**0.046206**	29	130.8	31.8	194.3	‡	‡
	12	**0.042362**	**0.042388**	24	14,635.3	38.6	14,711.9	0.000087	1.029913
	22	**0.008125**	0.008222	21	13,126.3	58.9	13,244.2	0	‡
	28	0.003824	0.014749	3	26,857.3	9.6	26,876.2†	‡	‡
	29	0.000042	0.024956	2	18,327.3	6.2	18,339.5†	0	‡
10	17	**0.078447**	**0.078531**	41	3,583.0	133.8	3,847.3	0	‡
	22	0.042638	0.064827	8	27,533.7	34.2	27,602.0†	‡	‡
	23	**0.036759**	0.074193	5	25,162.8	20.6	25,203.1†	0	‡

In addition, PWL functions requiring less breakpoints are computed for 4 instances (function 8, 9 and 10, all for $\delta = 0.05$ and 9 for $\delta = 0.005$ in addition), than those reported in Rebennack and Kallrath (2015). Optimal PWL functions for function 8 with 6 breakpoints and function 9 with 10 breakpoints are shown in Table 9 below. For function 9 and $\delta = 0.005$, Algorithm 1 finds a solution using 28 breakpoints with maximal difference of 0.0049851 after 40 iterations with a lower bound of 0.0049085 in 83,457.8 seconds. This

PWL function is shown in Fig. 3(f). Similarly, for function 10 and $\delta = 0.050$, the solution with maximal difference of 0.0472206 after 16 iterations requiring 10,486.5 seconds is computed with a lower bound of 0.0424170. The resulting PWL function is drawn in Fig. 3(g).

Particularly remarkable are the results for functions 7-10, where the GO approach can only yield loose bounds for B^*. This is explained by the rather involved functional forms of $f(x)$ which challenge global optimization software. The MILP approach proposed in this paper is only indirectly affected by the form of $f(x)$.

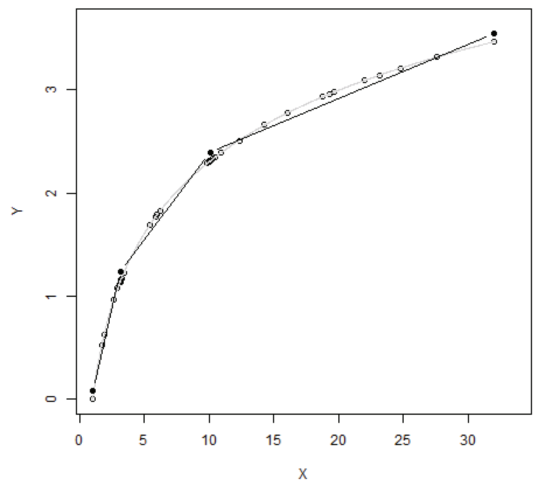
Table 9 summarizes the computational results for the maximal difference for the different test functions and different number of breakpoints. The computational budget was set to 5 hours. E_-^{B} reports upon the lower bound on the minimax difference E^{B}, while E_+^{B} is the upper bound. We stop Algorithm 1 if a difference between upper and lower bound of at most 0.0001 is obtained or the time limit, for each single optimization problem, is reached.

For functions 7-10, the MILP approach of this paper computed significantly tighter bounds on E^{B} for almost all instances compared to the GO approach. For 10 instances, where the GO approach cannot improve upon the trivial lower bound of 0, the MILP approach computed good lower bounds. An optimal E^{B}, for a tolerance of 0.0001, was computed for all 6 instances where the GO approach was able to compute it. For additional 13 instances, the MILP approach can compute optimal PWL functions, minimizing the maximal difference!

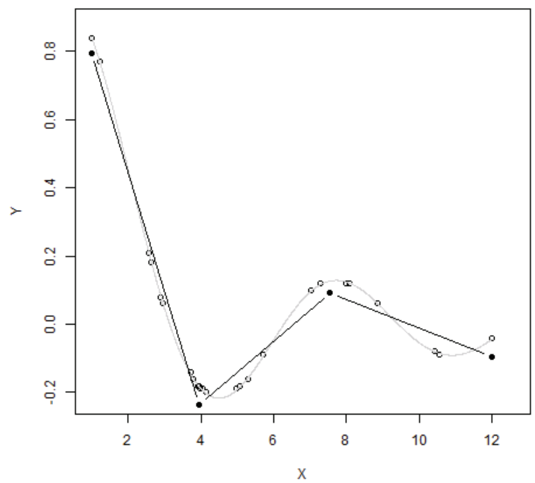
The plots in Fig. 3 show computed PWL functions, for a given number of breakpoints B. The black points mark the data points to approximate the continuous function f. Algorithm 1 adds these data points dynamically at critical places. It is clearly visible that the data points are not equidistantly distributed in the interval $[\mathbf{X}_-, \mathbf{X}_+]$. Also, with increasing number of breakpoints, more data points are required (this is mostly explained by the decrease in the maximal difference).

6. Conclusions

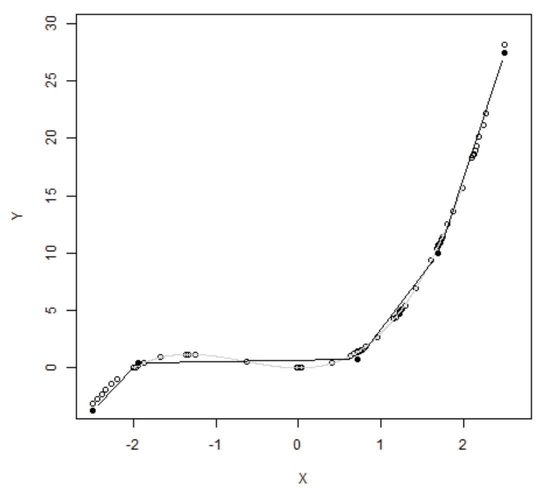
We present the first convex model formulations for globally optimal continuous univariate piecewise linear (PWL) function fitting. The non-convex constraints typically present in



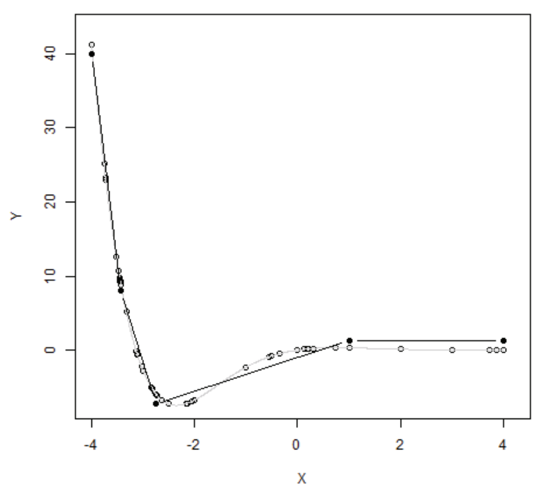
(a) #2, B=4



(b) #5, B=4



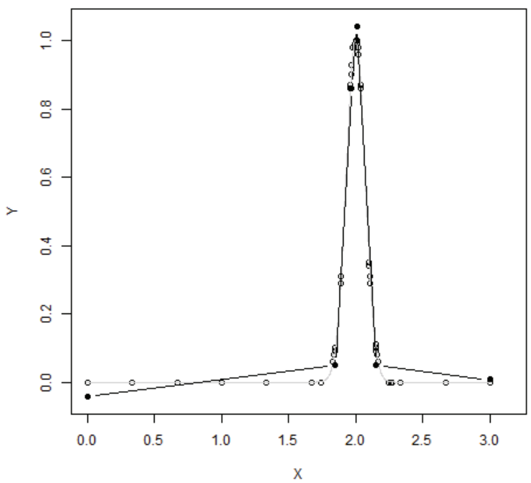
(c) #6, B=5



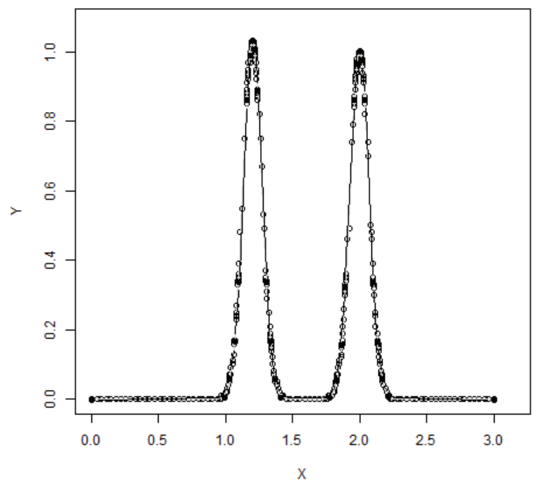
(d) #7, B=5

Figure 3 PWL functions for continuous $f(x)$. Function f is plotted in gray; the PWL function is black.

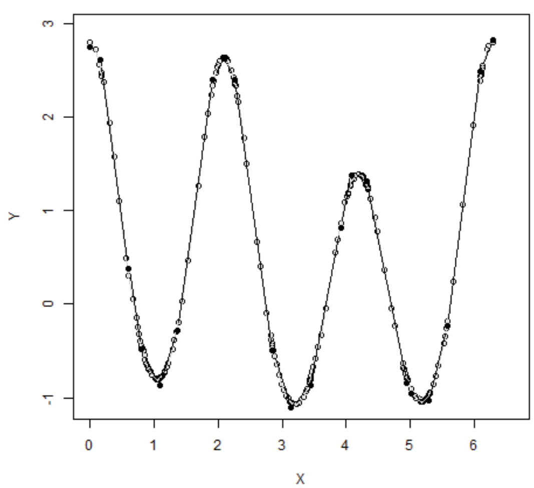
existing approaches stem from the continuity constraints, ensuring that two consecutive linear segments have the same value at each breakpoint. We avoid the function evaluation of the PWL function at these breakpoints. Instead, we propose a set of linear constraints,



(e) #8, B=6



(f) #9, B=28



(g) #10, B=21

Figure 3 continued.

together with additional binary variables, to indirectly enforce the continuity. The resulting continuous PWL functions are optimal with respect to a chosen metric among the sum of absolute differences, the sum of squared differences or the maximum difference. The proposed models are either MILPs or convex MIQPs, dependent on the metrics chosen. Part of the solution of these models is the optimal distribution of the breakpoints. When fitting convex PWL functions, the proposed models reduce to formulations known in the literature.

The proposed convex models allow the computation of continuous PWL functions to a set of discrete data. We further extend this model to also fit continuous univariate functions. Computational results for both benchmark instances for discrete data fitting and continuous function fitting demonstrate the superiority of the convex models. For discrete data fitting, a decrease in computational time of one order of magnitude is often observed. For continuous function fitting, the computational gains are even greater; several open fitting problems from the literature are solved.

Future work can investigate the role of the parameters $[\underline{C}, \bar{C}]$ and $[\underline{D}, \bar{D}]$ on the solution quality of the computed PWL functions. Of particular interest are safe choices of $[\underline{C}, \bar{C}]$ and $[\underline{D}, \bar{D}]$ for a given function f to allow the computation of a best possible PWL function.

Acknowledgments

This work was partially supported by the National Science Foundation Graduate Research Fellowship grant DGE-1057607.

References

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

Bellman, R., R. Roth. 1969. Curve fitting by segmented straight lines. *Journal of the American Statistical Association* **64** 1079–1084.

Bertsimas, D., A. King. 2016. OR Forum–an algorithmic approach to linear regression. *Operations Research* **64** 2–16.

Bertsimas, D., A. King, R. Mazumder. 2016. Best subset selection via a modern optimization lens. *The Annals of Statistics* **44** 813–852.

Bertsimas, D., R. Mazumder. 2014. Least quantile regression via modern optimization. *The Annals of Statistics* **42** 2494–2525.

Bertsimas, D., R. Shioda. 2007. Classification and regression via integer optimization. *Operations Research* **55** 252–271.

Chen, D. Z., H. Wang. 2009. *Algorithms and Computation: 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*, chap. Approximating Points by a Piecewise Linear Function: I. Springer Berlin Heidelberg, 224–233.

Croxton, K. L., B. Gendron, T. L. Magnanti. 2003. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science* **49** 1268–1273.

De Boor, C., J. R. Rice. 1968. Least squares cubic spline approximation, II-variable knots. Tech. rep., Computer Science Technical Reports, Purdue University.

Ertel, J. E., E. B. Fowlkes. 1976. Some algorithms for linear spline and piecewise multiple linear regression. *Journal of the American Statistical Association* **71** 640–648.

Feijoo, B., R. R. Meyer. 1988. Piecewise-linear approximation methods for nonseparable convex optimization. *Management Science* **34** 411–419.

Geißler, B. 2011. Towards Globally Optimal Solutions for MINLPs by Discretization Techniques with Applications in Gas Network Optimization. Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen-Nürnberg, Germany.

Geißler, B., A. Martin, A. Morsi, L. Schewe. 2012. Using Piecewise Linear Functions for Solving MINLPs. J. Lee, S. Leyffer, eds., *Mixed Integer Nonlinear Programming*, *The IMA Volumes in Mathematics and its Applications*, vol. 154. Springer, 287–314.

Goldberg, N., Y. Kim, S. Leyffer, T. D. Veselka. 2014. Adaptively refined dynamic program for linear spline regression. *Computational Optimization and Applications* **58** 523–541.

Guthery, S. B. 1974. Partition regression. *Journal of the American Statistical Association* **69** 945–947.

Hamann, B., J.-L. Chen. 1994. Data point selection for piecewise linear curve approximation. *Computer Aided Geometric Design* **11** 289–301.

Jupp, D. L. B. 1978. Approximation to data by splines with free knots. *SIAM Journal on Numerical Analysis* **15** 328–343.

Kallrath, J., S. Rebennack. 2014. Computing area-tight piecewise linear overestimators, underestimators and tubes for univariate functions. S. Butenko, C.A. Floudas, T.M. Rassias, eds., *Optimization in Science and Engineering*. Springer, 273–292.

Lohmann, T., S. Rebennack. 2017. Tailored Benders decomposition for a long-term power expansion model with short-term demand response. *Management Science* **63** 2027–2048.

MacGregor, J. F., T. J. Harris. 1993. The exponentially weighted moving variance. *Journal of Quality Technology* **25** 106–118.

Magnani, A., S. P. Boyd. 2009. Convex piecewise-linear fitting. *Optimization and Engineering* **10** 1–17.

Maranas, C., C. A. Floudas. 1994. Global Minimum Potential Energy Conformations of Small Molecules. *Journal of Global Optimization* **4** 135–170.

McCoy, K., V. Krasko, P. Santi, D. Kaffine, S. Rebennack. 2016. Minimizing economic impacts from post-fire debris flows in the Western United States. *Natural Hazards* **83** 149–176.

McGee, V. E., W. T. Carleton. 1970. Piecewise regression. *Journal of the American Statistical Association* **65** 1109–1124.

Misener, R., C. A. Floudas. 2013. GloMIQO: Global mixed-integer quadratic optimizer. *Journal of Global Optimization* **57** 3–50.

Muggeo, V. M. R. 2003. Estimating regression models with unknown break-points. *Statistics in Medicine* **22** 3055–3071.

Padberg, M. 2000. Approximating separable nonlinear functions via mixed zero-one programs. *Operations Research Letters* **27** 1–5.

Rebennack, S. 2016a. Combining sampling-based and scenario-based nested benders decomposition methods: application to stochastic dual dynamic programming. *Mathematical Programming* **156** 343–389.

Rebennack, S. 2016b. Computing tight bounds via piecewise linear functions through the example of circle cutting problems. *Mathematical Methods of Operations Research* **84** 3–57.

Rebennack, S. 2016c. Piecewise linear functions. *OR News* **58**.

Rebennack, S., J. Kallrath. 2015. Continuous Piecewise Linear Delta-Approximations for Univariate Functions: Computing Minimal Breakpoint Systems. *Journal of Optimization Theory and Applications* **167** 617–643.

Rosen, J. B., P. M. Pardalos. 1986. Global minimization of large-scale constrained concave quadratic problems by separable programming. *Mathematical Programming* **34** 163–174.

Schrage, L. 2004. LindoSystems: LindoAPI.

Sherali, H. D. 2001. On mixed-integer zero-one representations for separable lower-semicontinuous piecewise-linear functions. *Operations Research Letters* **28** 155–160.

Tawarmalani, M., N. V. Sahinidis. 2005. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming* **103** 225–249.

Toriello, A., J. P. Vielma. 2012. Fitting piecewise linear continuous functions. *European Journal of Operational Research* **219** 86–95.

Vielma, J. P., S. Ahmed, G. Nemhauser. 2010. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research* **58** 303–315.

Vielma, J. P., G. L. Nemhauser. 2011. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming* **128** 49–72.

Williams, C. M. 1978. An efficient algorithm for the piecewise linear approximation of planar curves. *Computer Graphics and Image Processing* **8** 286–293.

Wood, S. N. 2001. Minimizing model fitting objectives that contain spurious local minima by bootstrap restarting. *Biometrics* **57** 240–244.