



INFORMS Journal on Optimization

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Tightening the Difference-of-Convex Formulation for the Piecewise Linear Approximation in General Dimensions

Quentin Ploussard, Xiang Li, Matija Pavičević

To cite this article:

Quentin Ploussard, Xiang Li, Matija Pavičević (2025) Tightening the Difference-of-Convex Formulation for the Piecewise Linear Approximation in General Dimensions. INFORMS Journal on Optimization

Published online in Articles in Advance 29 Dec 2025

<https://doi.org/10.1287/ijoo.2025.0074>

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. You are free to download this work and share with others, but cannot change in any way or use commercially without permission, and you must attribute this work as “*INFORMS Journal on Optimization*. Copyright © 2025 The Author(s). <https://doi.org/10.1287/ijoo.2025.0074>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-nc-nd/4.0/>.”

Copyright © 2025 The Author(s)

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Tightening the Difference-of-Convex Formulation for the Piecewise Linear Approximation in General Dimensions

 Quentin Ploussard,^{a,*} Xiang Li,^b Matija Pavičević^a
^aArgonne National Laboratory, Lemont, Illinois 60439; ^bZhejiang University, Hangzhou 310027, China

*Corresponding author

Contact: qploussard@anl.gov,  <https://orcid.org/0000-0001-8209-5469> (QP); li.xiang@zju.edu.cn (XL); mpavicevic@anl.gov,

 <https://orcid.org/0000-0002-6524-5581> (MP)

Received: April 3, 2025


Revised: October 17, 2025;
November 25, 2025

Accepted: December 5, 2025

Published Online in *Articles in Advance*:
December 29, 2025

<https://doi.org/10.1287/ijoo.2025.0074>
Copyright: © 2025 The Author(s)

Abstract. This paper addresses the problem of tightening the mixed-integer linear programming (MILP) formulation for continuous piecewise linear (CPWL) approximations of data sets in arbitrary dimensions. The MILP formulation leverages the difference-of-convex (DC) representation of CPWL functions. We introduce the concept of well-behaved CPWL interpolations and demonstrate that any CPWL interpolation of a data set has a well-behaved version. This result is critical to tighten the MILP problem. We present six different strategies to tighten the problem, which include fixing the values of some variables, introducing additional constraints, identifying small big-M parameter values, and applying tighter variable bounds. These methods leverage key aspects of the DC representation and the inherent structure of well-behaved CPWL interpolations. Experimental results demonstrate that specific combinations of these tightening strategies lead to significant improvement in solution times, especially for tightening strategies that consider well-behaved CPWL solutions.

 **Open Access Statement:** This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. You are free to download this work and share with others, but cannot change in any way or use commercially without permission, and you must attribute this work as “*INFORMS Journal on Optimization*. Copyright © 2025 The Author(s). <https://doi.org/10.1287/ijoo.2025.0074>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-nc-nd/4.0/>.”

Funding: Financial support from the Department of Energy Office of Energy Efficiency and Renewable Energy [Contract DE-AC02-06CH11357] is gratefully acknowledged.

Supplemental Material: The electronic companion is available at <https://doi.org/10.1287/ijoo.2025.0074>.

Keywords: integer programming • linear programming • piecewise linear approximation

1. Introduction

A continuous piecewise linear (CPWL) function is a continuous function defined on a compact set which can be partitioned into a finite set of affine domains within which the function is affine. CPWL functions are a fundamental tool in mathematics, engineering, and data science. This is due to their ability to approximate any continuous nonlinear function defined on a compact set to an arbitrary level of precision using a finite number of linear pieces (Huang 2020). CPWL functions are also an essential tool when fitting a data set for which the analytical expression of the nonlinear relationship is unknown (Rebennack and Krasko 2020). In optimization models, using CPWL functions to represent complex nonlinear relationships enables the use of mature mixed-integer linear programming (MILP) solvers, which are often significantly faster than nonlinear solvers (Vielma et al. 2010, Vielma 2015).

Not only can CPWL functions be represented in MILP models, but the optimal CPWL fitting problem itself can also be expressed as a MILP problem (Toriello and Vielma 2012). The objective function may vary depending on the application. For instance, given a fixed number of affine functions, one may wish to determine the CPWL approximation that minimizes an approximation error metric—for example, the maximum or average error (Rebennack and Krasko 2020). Alternatively, one may wish to identify the CPWL approximation with the minimal number of affine pieces (Kong and Maravelias 2020).

MILP formulations for the CPWL fitting problem have been extensively investigated for univariate functions and one-dimensional data sets. Frenzen et al. (2010) studied optimal partitioning of the range of the univariate function to be approximated and provided a quantitative result regarding how the number of segments depends on the function and the approximation error. Rebennack and Krasko (2020) developed a finitely convergent

method to generate CPWL approximations for one-dimensional data sets or continuous univariate functions identifying the minimal number of breakpoints required to meet an acceptable approximation error. Similarly, Kong and Maravelias (2020) introduced a novel method of enforcing continuity at breakpoints through a set of linear constraints. The two formulations are compared in detail in Warwicker and Rebennack (2022), and it is shown that the formulation from Rebennack and Krasko (2020) is generally faster in practice. The work from Warwicker and Rebennack (2023) extends the work from Warwicker and Rebennack (2022) and uses a Benders decomposition approach to improve the solution time of the MILP problem. Ploussard (2024) introduces a tightening method that significantly shortens the MILP solution time of the CPWL fitting problem when hierarchically minimizing the number of linear pieces and approximation error. More recently, Warwicker and Rebennack (2025) presented an efficient approach for approximating continuous univariate functions with a CPWL function by embedding a linear-time algorithm into an adaptive discretization framework.

For bivariate functions and two-dimensional data sets, most existing methods constrain the affine domains of the CPWL approximation to be simplices. For example, D'Ambrosio et al. (2010) partition the CPWL function domain into rectangles that are later split into triangles along predefined diagonals. Toriello and Vielma (2012) use a similar method but select a set of splitting diagonals that minimize the approximation error. Rebennack and Kallrath (2015) use a different triangulation strategy that iteratively partitions the triangles covering the approximation domain into smaller ones until a target approximation error is reached. Duguet and Ngueveu (2022) propose a similar iterative refinement, but without imposing the affine domains to be simplices. Bärmann et al. (2023) introduce an approximation algorithm for the optimal piecewise linear interpolation of the continuous bivariate function xy over rectangular domains.

The literature addressing the optimal CPWL fitting problem for data sets of higher dimensions is significantly more limited (Misener and Floudas 2010, Rebennack and Kallrath 2015). Besides, most of the existing methods essentially extend the simplex-based partitioning strategy used for two dimensions, which result in computational challenges due to the exponential growth of the number of possible simplex-based partitions as the data dimension increases (Hughes and Anderson 1996). Kazda and Li (2021) introduced the first MILP formulation able to identify the optimal CPWL approximation of data sets of any dimension for a given maximum approximation error. They achieve this by leveraging the difference-of-convex (DC) representation of CPWL functions (Kripfganz and Schulze 1987), which allows for the affine domains partitioning the CPWL function to be implicitly defined by the difference of the pointwise maxima of two sets of affine functions. The strength of the DC representation lies in its ability to describe any CPWL function with affine domains of arbitrary shape. However, the computational cost of the MILP problem rapidly increases with the dimension of the data set, the number of data points, and the number of affine functions.

Deep learning techniques offer an alternative way to identify CPWL approximations (Huang 2020). In fact, the function represented by a neural network (NN) using ReLU activation function is a CPWL function, and the NN loss function represents the CPWL approximation error (DeVore et al. 2021). To the best of our knowledge, the only existing methods to identify CPWL approximations of data sets in multiple dimensions without explicitly specifying the partitioning of affine domains are the ReLU NN approach and the DC MILP approach introduced by Kazda and Li (2021). Training an NN to identify a good CPWL approximation is significantly faster than solving an MILP problem, contributing to NN growing popularity in recent years. However, the MILP approach provides three key advantages over the ReLU NN approach:

- The NN approach uses a gradient-based method that identifies local optima, whereas the MILP approach is guaranteed to yield the CPWL function with the smallest approximation error.
- Contrary to the MILP approach, the NN approach does not allow for the explicit enforcement of a target maximum approximation error.
- Because of the complex relationship between the structure (width and depth) of an NN and the number of affine pieces of the CPWL functions they can model, it is difficult for NNs to identify good CPWL approximations with a manageable number of affine pieces (Chen et al. 2023). Conversely, the MILP approach allows the user to identify CPWL functions with a predefined, or minimal, number of linear pieces. This is a critical feature when these CPWL functions are subsequently embedded in MILP formulations to model nonlinear relationships, where the number of binary variables required to model CPWL functions typically increases with the number of affine pieces that compose them, making the MILP problem harder to solve. In other words, although an NN can quickly identify a good-quality CPWL approximation of a nonlinear relationship, the resulting approximation may render the MILP formulation in which it is embedded computationally intractable.

To address the high computational cost of solving the MILP problem, authors Kazda and Li (2024) introduce an iterative linear programming (LP) approach that identifies a valid CPWL approximation given a target maximum error. However, the identified CPWL solution is not necessarily optimal. A priori, identifying the global

optimum of the CPWL fitting problem can only be achieved by solving an MILP problem. Therefore, it is critical to improve the MILP formulation of the CPWL fitting problem to reduce its computational cost. The method described in Ploussard (2024) achieves a tightening of the MILP problem that significantly improves the solution time, but the method only applies to one-dimensional data sets. This raises the need to develop MILP tightening methods that can be extended to data sets of any dimensions.

This paper aims to address this research gap. To achieve this, we introduce a critical class of CPWL interpolations called “well-behaved” and demonstrate that any CPWL interpolation has a well-behaved version. This important result is then leveraged to identify six tightening strategies that can be combined to significantly reduce the solution time of the MILP CPWL fitting problem.

The five main contributions of this paper are the following. First, we formalize the concept of well-behaved CPWL interpolations, a class of CPWL interpolations in which each affine piece interpolates a number of points greater than the dimension of the interpolated data set. Second, we demonstrate that any CPWL interpolation of a data set has a well-behaved version, which serves as a key theoretical foundation for our tightening method. Third, we leverage this result to introduce six tightening strategies for the MILP formulation of the CPWL fitting problem. Fourth, we analyze the theoretical impact of each tightening strategy together with the time complexity of the preprocessing step involved. Fifth, we assess the practical impact of multiple combinations of the tightening strategies. The proposed tightening strategies enable the identification of an optimal CPWL approximation in a relatively low computation time (i.e., up to 20 times faster than the existing literature).

The remainder of the article is organized as follows: Section 2 introduces the theoretical background of the CPWL-fitting problem. Section 3 describes the proposed MILP formulation. Section 4 formalizes the concept of well-behaved CPWL fitting and demonstrates the fact that any CPWL interpolation or approximation of a data set has a well-behaved version. Section 5 introduces the six tightening strategies of the MILP formulation. Section 6 introduces the case studies and assesses the performances of several combinations of tightening strategies. Section 7 presents the conclusions.

2. Background

Let $S = (x_i, z_i)_{i=1, \dots, N} = (x_{i,1}, \dots, x_{i,d}, z_i)_{i=1, \dots, N}$ be a set of N points of \mathbb{R}^{d+1} . We assume that the points $(x_i)_{i=1, \dots, N}$ are in general position in \mathbb{R}^d —that is, no subset of $d + 1$ points is contained within a single hyperplane (Edelsbrunner et al. 1986). Let $P_{\mathbb{R}^d} : \mathbb{R}^{d+1} \rightarrow \mathbb{R}^d$ be the projection defined as $P_{\mathbb{R}^d}(x_1, \dots, x_d, z) = (x_1, \dots, x_d)$. Let $S_{\mathbb{R}^d} = (x_i)_{i=1, \dots, N}$ be the projection of S by $P_{\mathbb{R}^d}$. Let $D = \text{Conv}(S_{\mathbb{R}^d})$ be the convex hull of $S_{\mathbb{R}^d}$.

Definition 1. We say that $f : D \rightarrow \mathbb{R}$ is a CPWL function when f is continuous and there is a set of P affine functions $\{f_k : D_k \rightarrow \mathbb{R}, k = 1, \dots, P\}$ such that the D_k are convex polytopes, $\cup_{k=1}^P D_k = D$, and $f|_{D_k} = f_k, \forall k \in \{1, \dots, P\}$. The f_k are called the affine pieces of f , and the D_k are called the affine domains of f .

Note that, contrary to the more common definition of CPWL functions from Geißler et al. (2012), we do not assume that the D_k form a partition of D . The intersection of two affine domains is not required to have an empty interior. In addition, an affine domain can be empty, and two affine domains can be equal as long as the corresponding affine pieces are equal. This implies that the set of affine functions representing the CPWL function is not unique. However, apart from this, Definition 1 describes exactly the same mathematical object as the CPWL functions defined in the literature. The flexibility of our definition enables us to compare CPWL functions to one another, which will be useful in the following section.

Proposition 1. Let $f : D \rightarrow \mathbb{R}$ be a CPWL function. f can be expressed as the difference of two convex CPWL functions f^+ and f^- . Specifically, there is a set of P^+ affine functions $\{f_j^+ : D \rightarrow \mathbb{R}, j \in \{1, \dots, P^+\}\}$ and P^- affine functions $\{f_k^- : D \rightarrow \mathbb{R}, k \in \{1, \dots, P^-\}\}$ such that:

$$f(x) = f^+(x) - f^-(x) = \max_{j \in \{1, \dots, P^+\}} f_j^+(x) - \max_{k \in \{1, \dots, P^-\}} f_k^-(x), \forall x \in D.$$

f_j^c are called the affine pieces of f^c , and $D_j^c = \{x \in D : f^c(x) = f_j^c(x)\}$ are called the affine domains of $f^c, j \in \{1, \dots, P^c\}, c \in \{+, -\}$. In addition, $\{D_j^c : j \in \{1, \dots, P^c\}\}$ is a partition of D .

The proof of Proposition 1 is provided in Kripfganz and Schulze (1987). The above formulation is called the DC representation of f . Note that the DC representation of a CPWL function is not unique.

Proposition 2. Let $f : D \rightarrow \mathbb{R}$ be a CPWL function. Let $f = f^+ - f^-$ be a DC representation of f . The affine domains of f^+ and f^- are convex polytopes. In addition, any affine domain of f is the intersection of an affine domain of f^+ and f^- . In other

words, for any affine domain D_p of f , there is an affine domain D_j^+ and D_k^- of f^+ and f^- such that $D_p = D_j^+ \cap D_k^-$, and $f(x) = f_j^+(x) - f_k^-(x)$, $\forall x \in D_p$.

The proof of Proposition 2 is given in Kripfganz and Schulze (1987) and stems from the DC representation of f .

Definition 2. We say that $f : D \rightarrow \mathbb{R}$ is an interpolation of, or interpolates, S when $f(x_i) = z_i$, $\forall i \in \{1, \dots, N\}$ (Davis 1975).

Definition 3. We say that $f : D \rightarrow \mathbb{R}$ is an ε -approximation of, or ε -approximates, S when there exists $(e_i)_{i=1, \dots, N} \in \mathbb{R}^N$ such that $|e_i| \leq \varepsilon$ and f interpolates the set $(x_i, z_i + e_i)_{i=1, \dots, N}$. In other words, f is an ε -approximation of S if $\max_{i=1, \dots, N} |f(x_i) - z_i| \leq \varepsilon$.

Definition 4. Let $f : D \rightarrow \mathbb{R}$ and $g : D \rightarrow \mathbb{R}$ be two ε -approximations of S . We say that f and g are equivalent with respect to S when $f(x_i) = g(x_i)$, $\forall i \in \{1, \dots, N\}$.

3. MILP Formulation

We aim to identify the optimal CPWL fitting of a set of points S . More specifically, we aim to identify the CPWL ε -approximation of S that minimizes an objective Q subject to a given error tolerance ε . For example, Q can be the maximum fitting error, the average fitting error, or the number of affine pieces.

The following MILP formulation is largely based on the formulation presented in Kazda and Li (2021), where $f(x_i)$, $f^+(x_i)$, $f^-(x_i)$, e_i , a_j^c , and b_j^c are continuous variables; $\delta_{i,j}^c$ are binary variables; and M_i^c are big-M parameters.

$$\text{minimize } Q, \tag{1}$$

$$\text{s.t. } f(x_i) = f^+(x_i) - f^-(x_i), \quad i \in \{1, \dots, N\}, \tag{2}$$

$$0 \leq f^c(x_i) - a_j^{cT} x_i - b_j^c, \quad i \in \{1, \dots, N\}, \quad j \in \{1, \dots, P^c\}, \quad c \in \{+, -\}, \tag{3}$$

$$f^c(x_i) - a_j^{cT} x_i - b_j^c \leq M_i^c(1 - \delta_{i,j}^c), \quad i \in \{1, \dots, N\}, \quad j \in \{1, \dots, P^c\}, \quad c \in \{+, -\}, \tag{4}$$

$$\sum_{j=1}^{P^c} \delta_{i,j}^c \geq 1, \quad i \in \{1, \dots, N\}, \quad c \in \{+, -\}, \tag{5}$$

$$-e_i \leq f(x_i) - z_i \leq e_i, \quad i \in \{1, \dots, N\}, \tag{6}$$

$$e_i \leq \varepsilon, \quad i \in \{1, \dots, N\}, \tag{7}$$

$$\delta_{i,j}^c \in \{0, 1\}, \quad i \in \{1, \dots, N\}, \quad j \in \{1, \dots, P^c\}, \quad c \in \{+, -\}, \tag{8}$$

$$f(x_i), f^+(x_i), f^-(x_i) \in \mathbb{R}, \quad e_i \in \mathbb{R}_{\geq 0}, \quad i \in \{1, \dots, N\}, \tag{9}$$

$$a_j^c \in \mathbb{R}^d, \quad b_j^c \in \mathbb{R}, \quad j \in \{1, \dots, P^c\}, \quad c \in \{+, -\}. \tag{10}$$

Equation (1) represents the objective function to be minimized. As long as Q is a linear expression, Equations (1)–(10) formulate an MILP problem. For the rest of the paper, we will refer to this MILP problem as $MILP1(Q)$. Equation (2) is the DC representation of the CPWL function, where $f(x_i)$, $f^+(x_i)$, and $f^-(x_i)$ represent the value at x_i of the functions f , f^+ , and f^- . Equations (3) and (4) formulate the representation of each convex function f^c as the maximum of a set of affine functions f_j^c , where a_j^c and b_j^c are the linear coefficients and bias terms of each affine piece f_j^c . The binary variable $\delta_{i,j}^c$ is equal to one when the point x_i belongs to the domain D_j^c , and the big-M parameter M_i^c ensures that $f^c(x_i) = f_j^c(x_i)$ when $x_i \in D_j^c$. Note that the big-M parameter M_i^c in Equation (4) must be large enough so that it does not constrain the term $f^c(x_i) - a_j^{cT} x_i - b_j^c$ when $\delta_{i,j}^c = 0$. In Equation (5), we will identify a tight value for M_i^c . P^+ and P^- are the numbers of affine pieces of f^+ and f^- , respectively. Equation (5) ensures that each point x_i belongs to at least one affine domain of f^+ and one affine domain of f^- . Note that, contrary to the literature (e.g., Kazda and Li 2021), we use an inequality constraint instead of an equality constraint. By doing so, we allow the points x_i to belong to multiple affine domains. This can occur if, for example, some affine pieces are identical or if some points are at the border between two or more affine domains. This inequality is also critical to ensure that some well-behaved CPWL solutions are feasible in $MILP1(Q)$, which is a property that will be discussed in the next section. Equation (6) defines the approximation error e_i at each point x_i as the distance between z_i and the value of the CPWL function at x_i . Equation (7) ensures that this distance is never greater than the specified error tolerance ε . Finally, Equations (8)–(10) define the domain of the continuous and binary variables.

Note that some optimization solvers like Gurobi (Gurobi Optimization LLC 2024a) make it possible to formulate Equation (4) without using a big-M parameter. Instead, Equation (4) can be replaced by the indicator

constraint below (Gurobi Optimization LLC 2024b):

$$\delta_{i,j}^c = 1 \Rightarrow f^c(x_i) - a_j^{cT} x_i - b_j^c \leq 0, \quad i \in \{1, \dots, N\}, \quad j \in \{1, \dots, P^c\}, \quad c \in \{+, -\}. \quad (11)$$

We note $MILPIC1(Q)$ the alternative MILP formulation using the indicator constraint, which is composed of Equations (1)–(3) and (5)–(11).

The objective can be set to minimize different error metrics. Formulating the objective with Equation (12) minimizes the average fitting error (L_1 norm), the discrete analogue of minimizing the area between two functions. Alternatively, using Equation (13) minimizes the maximum fitting error (L_∞ norm), which provides a strict point-wise guarantee on the approximation quality.

$$Q = \frac{1}{N} \sum_{i=1}^N e_i, \quad (12)$$

$$Q \geq e_i, \quad i \in 1, \dots, N. \quad (13)$$

Other objective functions may include the number of affine pieces of f^+ , f^- , or f , as well as a combination of error metric and number of affine pieces. These alternative objective functions can be found in Electronic Companion Section EC.1.

4. Well-Behaved CPWL Interpolations

This section introduces a new class of CPWL interpolations called “well-behaved” CPWL interpolations. Loosely speaking, well-behaved CPWL interpolations are composed of affine pieces whose gradients are not excessively steep. This class of CPWL interpolations aligns with the intuitive representation of what the CPWL interpolation of a data set should look like, and they have desirable properties when modeled in $MILP1(Q)$. This section formalizes the definition of a well-behaved CPWL interpolation and defines what constitutes a well-behaved version of a CPWL interpolation. We then demonstrate that any CPWL interpolation has a well-behaved version. This result is critical to the tightening procedure introduced in Section 5.

Proposition 3. Let $S = (x_i, z_i)_{i=1, \dots, d+1}$ be a set of $(d + 1)$ points of \mathbb{R}^{d+1} , where the x_i are in general position. There exists a unique linear interpolation of S . In other words, S can be interpolated by a unique affine function.

Proposition 3 is a fundamental result from linear algebra (Boyd and Vandenberghe 2004). The system of linear equations $a^T x_i + b = z_i, i \in \{1, \dots, d + 1\}$, with variables $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$, is exactly determined and has a unique solution due to the general position of the x_i . If the number of points is less than $d + 1$, the system is underdetermined, and there is an infinite number of possible linear interpolations. If the number of points is greater than $d + 1$, because the x_i are in general position, the system becomes overdetermined, and there exists either one or no linear interpolation.

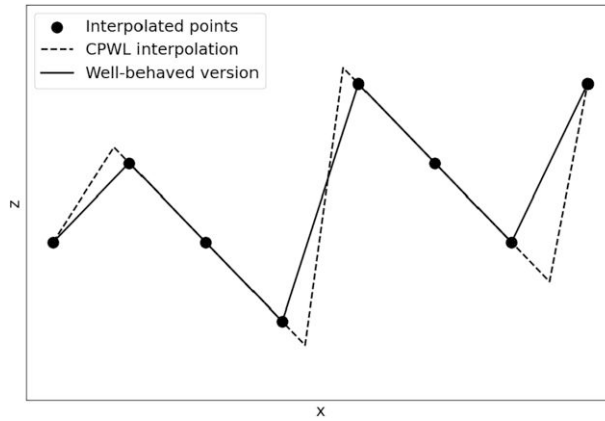
Definition 5. Let $f : D \rightarrow \mathbb{R}$ be a CPWL interpolation of S , with $|S| = N \geq d + 1$. We say that an affine piece of f is underdetermined, exactly determined, or overdetermined with respect to S if it interpolates less than, exactly, or more than $d + 1$ points of S , respectively.

Definition 6. We say that $f : D \rightarrow \mathbb{R}$ is a well-behaved CPWL interpolation of S when f is a CPWL interpolation of S and each affine piece of f is exactly determined or overdetermined—that is, each affine piece interpolates at least $d + 1$ points of S . Additionally, let $f : D \rightarrow \mathbb{R}$ be an ε -approximation of S and $e_i = f(x_i) - z_i$. We say that f is a well-behaved CPWL ε -approximation of S when f is a well-behaved CPWL interpolation of $(x_i, z_i + e_i)_{i=1, \dots, N}$.

Definition 7. Let f and g be two CPWL interpolations of S composed of the set of affine pieces $\{f_k : D_k \rightarrow \mathbb{R}, k = 1, \dots, P\}$ and $\{g_k : D'_k \rightarrow \mathbb{R}, k = 1, \dots, P\}$, respectively. We say that g is a well-behaved version of the CPWL interpolation f with respect to S if g is a well-behaved CPWL interpolation of S and $D_k \cap S_{\mathbb{R}^d} \subseteq D'_k \cap S_{\mathbb{R}^d}, \forall k \in \{1, \dots, P\}$. Additionally, let f and g be two CPWL ε -approximations of S and $e_i = f(x_i) - z_i$. We say that g is a well-behaved version of the CPWL ε -approximation f with respect to S if g is a well-behaved version of the CPWL interpolation f with respect to $(x_i, z_i + e_i)_{i=1, \dots, N}$.

Remark 1. If g is a well-behaved version of the CPWL ε -approximation f with respect to S , then the ε -approximations f and g are equivalent according to Definition 4.

In other words, a well-behaved version of a CPWL interpolation f of S is a transformation of f where each affine piece has been adjusted, or “tilted,” to interpolate at least $d + 1$ points of S , including the points it was originally interpolating (before being tilted). An illustration for $d = 1$ is provided in Figure 1. In the following sections, we

Figure 1. Illustration of a CPWL Interpolation and Its Well-Behaved Version for $d = 1$ 

Note. Both CPWL functions interpolate the points and are composed of five affine pieces, but the affine pieces of the well-behaved interpolation have been tilted to interpolate at least two points per piece.

show that well-behaved CPWL interpolations have interesting properties that can be used to tighten the MILP formulation. To the best of the authors' knowledge, this is the first time this class of CPWL interpolations is introduced. We prove below that for any CPWL interpolation, a well-behaved version always exists.

Remark 2. If f is well-behaved, then f is a well-behaved version of itself.

Lemma 1. Let S be a set of N points in \mathbb{R}^{d+1} , with $N \geq d + 1$. Let $f : D \rightarrow \mathbb{R}$ be a CPWL interpolation of S . Let $f_1 : D_1 \rightarrow \mathbb{R}$ be one of the affine pieces of f , and let $f_2 : D_2 \rightarrow \mathbb{R}$ be a neighboring affine piece of f_1 —that is, an affine piece of f sharing a common domain boundary with f_1 . Let \tilde{f}_1 be the affine extension of f_1 to the domain D . Then,

$$\exists c \in \{-1, 1\} : \forall x \in D_2, \quad c(\tilde{f}_1(x) - f_2(x)) \leq 0.$$

Proof of Lemma 1. Let $x \in D_2$. According to Proposition 2, a shared domain boundary between f_1 and f_2 is either a shared domain boundary between two affine pieces of f^+ or between two pieces of f^- .

Case 1: The shared domain boundary is between two affine pieces of f^+ . Then, using Proposition 2, we can write $\tilde{f}_1(x) = f_1^+(x) - f_0^-(x)$ and $f_2(x) = f_2^+(x) - f_0^-(x)$. Because $f_2^+(x) = \max_{j \in \{1, \dots, p^+\}} f_j^+(x) \geq f_1^+(x)$, we therefore have $\tilde{f}_1(x) \leq f_2(x)$. Thus $(\tilde{f}_1(x) - f_2(x)) \leq 0$.

Case 2: The shared domain boundary is between two affine pieces of f^- . We can write $\tilde{f}_1(x) = f_0^+(x) - f_1^-(x)$ and $f_2(x) = f_0^+(x) - f_2^-(x)$. Because $f_2^-(x) \geq f_1^-(x)$, we therefore have $\tilde{f}_1(x) \geq f_2(x)$. Thus, $(-1)(\tilde{f}_1(x) - f_2(x)) \leq 0$. \square

Proposition 4. Let $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Assume that $\min(n, m)$ rows of A are linearly independent and the polyhedron $U = \{x \in \mathbb{R}^n : Ax \leq \mathbf{b}\}$ is nonempty. Then, there exists a point $x^* \in \mathbb{R}^n$ such that $A_i x^* = b_i$ for $\min(n, m)$ rows A_i of A .

Proposition 4 is a well-established fact from LP theory, as described in Bertsimas and Tsitsiklis (1997). The $\min(n, m)$ constraints " $A_i x^* = b_i$ " are also known as the active constraints of the solution x^* . The point x^* lies on the boundary of U , and, if $m \geq n$, x^* is a vertex of U .

Lemma 2. Let S be a set of N points in \mathbb{R}^{d+1} , with $N \geq d + 1$. Let $f : D \rightarrow \mathbb{R}$ be a CPWL interpolation of S that is not well-behaved—that is, at least one of the affine pieces of f is underdetermined. Let $f_0 : D_0 \rightarrow \mathbb{R}$ represent an underdetermined affine piece of f that interpolates n points of S ($n \leq d$). Let $f_j : D_j \rightarrow \mathbb{R}, j \in \{1, \dots, J\}$ be the J neighboring pieces of f_0 —that is, the affine pieces that share a domain boundary with f_0 . Let $\tilde{D}_0 = \cup_{j=0}^J D_j$. Assume there is a total of $m \geq 0$ points in S whose projections lie in $\tilde{D}_0 \setminus D_0$ (interpolated by the J neighboring pieces but not by f_0). Then, there exists an alternative CPWL interpolation of S in which f_0 is adjusted to interpolate $\min(d + 1, n + m)$ points of S without impacting the points interpolated by the neighboring pieces. In other words, we can construct a new CPWL interpolation of S defined over \tilde{D}_0 and represented by a set of affine functions $\{f'_j : D'_j \rightarrow \mathbb{R}, j \in \{0, 1, \dots, J\}\}$, such that: (1) The functions f'_j and f_j are equal on their shared domain—that is, $f'_j(x) = f_j(x), \forall x \in D'_j \cap D_j, j \geq 1$; (2) the points that are interpolated by f_j and f'_j are the same—that is, $S_{\mathbb{R}^d} \cap D'_j = S_{\mathbb{R}^d} \cap D_j, j \geq 1$; and (3) the function f'_0 interpolates $\min(d + 1, n + m)$ points of S , which include the n points interpolated by f_0 —that is, $|S_{\mathbb{R}^d} \cap D'_0| = \min(d + 1, n + m)$ and $S_{\mathbb{R}^d} \cap D_0 \subseteq S_{\mathbb{R}^d} \cap D'_0$.

Proof of Lemma 2. Let \mathbf{a}_0 and b_0 be the linear coefficients and bias terms of f_0 —that is, $f_0(\mathbf{x}) = \mathbf{a}_0^T \mathbf{x} + b_0, \forall \mathbf{x} \in D_0$. Let $(x_i, z_i)_{i=1, \dots, n}$ denote the n points of S whose projections on \mathbb{R}^d lie in D_0 and $(x'_i, z'_i)_{i=1, \dots, m}$ denote the m points of S whose projections on \mathbb{R}^d lie in $\tilde{D}_0 \setminus D_0$. Let $p(i)$ denote the smallest integer such that $x'_i \in D_{p(i)}$. According to Lemma 1, $\exists (c_j)_{j=1, \dots, J} \in \{-1, 1\}^J : c_{p(i)}(\tilde{f}_0(x'_i) - f_{p(i)}(x'_i)) \leq 0, \forall i \in \{1, \dots, m\}$. Because $\tilde{f}_0(x'_i) = \mathbf{a}_0^T x'_i + b_0$ and $f_{p(i)}(x'_i) = z'_i$, this can be expressed as: $c_{p(i)}(\mathbf{a}_0^T x'_i + b_0 - z'_i) \leq 0, \forall i \in \{1, \dots, m\}$. In other words, (\mathbf{a}_0, b_0) is a point of the polyhedron U defined by:

$$\begin{aligned}
 &(\mathbf{a}, b) \in \mathbb{R}^{d+1} : \\
 &\mathbf{a}^T \mathbf{x}_i + b - z_i = 0, \quad i \in \{1, \dots, n\} \\
 &c_{p(i)}(\mathbf{a}^T \mathbf{x}'_i + b - z'_i) \leq 0, \quad i \in \{1, \dots, m\}.
 \end{aligned}$$

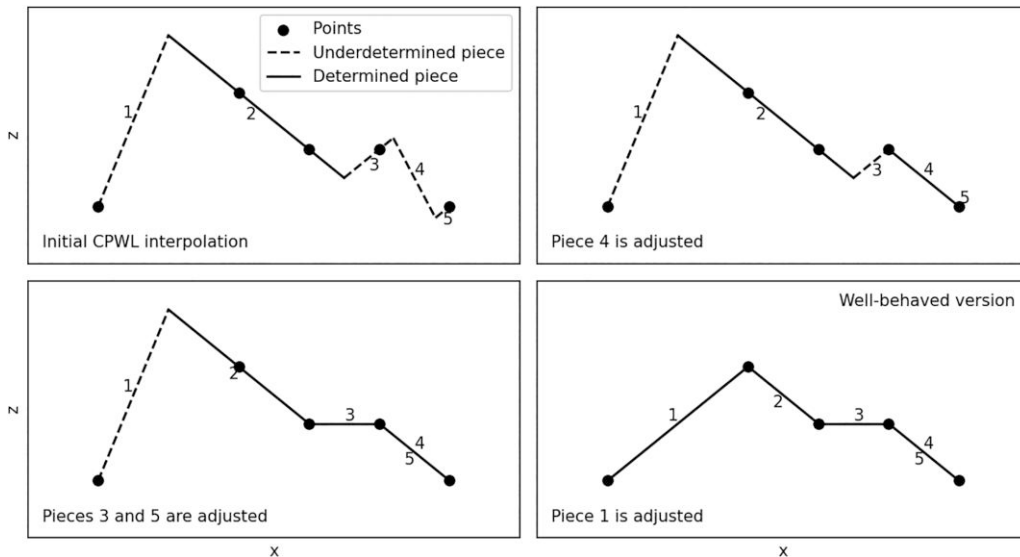
U is not empty because $(\mathbf{a}_0, b_0) \in U$. Therefore, according to Proposition 4, there exists a solution $(\mathbf{a}^*, b^*) \in U$ with $\min(d + 1, n + m)$ active constraints. Note that the constraints “ $\mathbf{a}^{*T} \mathbf{x}_i + b^* - z_i = 0$ ” are already active, which means that there must be $\min(d + 1, n + m) - n$ additional active constraints of type “ $c_{p(i)}(\mathbf{a}^{*T} \mathbf{x}'_i + b^* - z'_i) = 0$.” We define f'_0 as the affine function defined by $f'_0(\mathbf{x}) = \mathbf{a}^{*T} \mathbf{x} + b^*$. We define the domain of f'_0 as the domain delimited by the J boundaries $\{\mathbf{x} \in D : f'_0(\mathbf{x}) = \tilde{f}_j(\mathbf{x})\}$. We define f'_j as the J affine pieces f_j for which the domains were updated according to the J boundaries of f'_0 . The set of affine functions $\{f'_j : D'_j \rightarrow \mathbb{R}, j \in \{0, 1, \dots, J\}\}$ constitutes a valid CPWL interpolation of S over \tilde{D}_0 , with f'_0 interpolating $\min(d + 1, n + m)$ points of S , including the original n points interpolated by f_0 . The points of S interpolated by f'_j are the same as the points interpolated by $f_j, \forall j \in \{1, \dots, J\}$. \square

Theorem 1. Let S be a set of N points in \mathbb{R}^{d+1} , where $N \geq d + 1$. Any CPWL interpolation of S has a well-behaved version.

Proof of Theorem 1. Let $f : D \rightarrow \mathbb{R}$ be a CPWL interpolation of S . If every affine piece of f interpolates at least $d + 1$ points, f is already well-behaved. Otherwise, we construct a well-behaved version iteratively. The procedure begins by identifying an underdetermined piece, f_0 , with a neighboring domain containing at least one point not interpolated by f_0 ; the existence of such a piece is proven in Section EC.2 of the electronic companion. Using Lemma 2, we then adjust f_0 to interpolate this additional point without affecting its neighbors. Repeating this process guarantees termination, as each step incorporates a new point from the finite set S . The final CPWL interpolation is, by construction, a well-behaved version of f . Note that this constructive procedure serves as a theoretical proof of existence of the well-behaved version; a computational implementation is not required for the scope of this work. The process is illustrated in Figure 2. \square

Corollary 1. Any CPWL ε -approximation of S has a well-behaved version.

Figure 2. Illustration of the Process of Converting a CPWL Interpolation into Its Well-Behaved Version for $d = 1$



Notes. At the beginning of the process, piece 5 could not be selected because its neighboring piece (piece 4) does not interpolate any point. At the end of the process, pieces 4 and 5 have identical domain, which is allowed by our definition of CPWL functions.

The proof of Corollary 1 directly follows from Theorem 1 and Definition 7.

Remark 3. Note that the procedure of tilting the affine pieces of f to construct a well-behaved version may result in two or more underdetermined pieces merging together. In other words, initially distinct pieces may end up interpolating the same subset of points and having the same expression and affine domain after converting f to a well-behaved interpolation. In this case, the well-behaved version is still a valid CPWL function owing to the flexibility of Definition 1.

Remark 4. A well-behaved version of a CPWL interpolation is not necessarily unique. That is, a CPWL interpolation may have several well-behaved versions. This stems from the fact that the point with $\min(n, m)$ active constraints in Proposition 4 is not necessarily unique.

An important application of this theorem is that we can significantly reduce the set of CPWL functions to be considered in $MILP1(Q)$ without impacting the feasibility of the problem. Specifically, if a CPWL ε -approximation f of S exists, a well-behaved version g of f also exists. The two CPWL solutions f and g have the same number of affine pieces and are equivalent with respect to S —that is, $f(x_i) = g(x_i)$, $\forall i \in \{1, \dots, N\}$. Therefore, we can eliminate CPWL solutions that are not well-behaved from the feasible region of $MILP1(Q)$ because they are redundant solutions and may have excessively steep gradients due to their underdetermined affine pieces. In the next section, we will see how this theorem allows us to derive valid tight bounds and constraints for the MILP problem.

Definition 8. Herein, we will refer to the set of CPWL ε -approximations of S as $CPWL(S, \varepsilon)$. The subset of $CPWL(S, \varepsilon)$ that can be represented using P^+, P^- affine pieces for f^+, f^- , respectively, is denoted $CPWL(S, \varepsilon, P^+, P^-)$. The set of well-behaved CPWL ε -approximations of S will be denoted as $CPWL^*(S, \varepsilon)$, and $CPWL^*(S, \varepsilon, P^+, P^-)$ when represented by P^+, P^- affine pieces. Note that $CPWL^*(S, \varepsilon) \subset CPWL(S, \varepsilon)$. Additionally, note that the feasible region of the MILP problem $MILP1(Q)$ is given by $CPWL(S, \varepsilon, P^+, P^-)$.

5. Tightening the MILP Formulation

This section introduces six strategies to tighten and enhance the formulation of $MILP1(Q)$. First, we demonstrate that we can fix one of the affine pieces of f^- without affecting the set of CPWL solutions. Next, we impose an ordering on the affine pieces of f^+ and f^- that does not affect the set of CPWL solutions. We can also impose that each affine piece of f^+ and f^- contains at least $d + 1$ points, which serves as a valid tightening of the set of feasible well-behaved CPWL solutions. Alternatively, we may require that each affine piece of f contains at least $d + 1$ points, which requires the use of additional variables. Finally, we identify tight values for the big-M parameters and establish tight bounds for the linear variables.

Additional strategies that leverage the convexity of the functions f^+ and f^- and of their affine domains are included in the electronic companion, Section EC.3. However, contrary to the six strategies presented here, their impact on the feasible region is unclear, and they do not seem to have a significant impact on the solution time.

Herein, it is assumed that $N \geq d + 1$.

5.1. Fixing One Affine Piece

Lemma 3. Let $\{f_k, k \in \{0, 1, \dots, K\}\}$ be a set of functions. We have $\max_{k \in \{1, \dots, K\}} (f_k - f_0) = \max_{k \in \{1, \dots, K\}} (f_k) - f_0$.

Lemma 3 follows directly from the translation invariance of the max function—that is, $\max(a - c, b - c) = \max(a, b) - c$, $\forall a, b, c \in \mathbb{R}$.

Theorem 2. The following equation constitutes a valid tightening constraint of $MILP1(Q)$:

$$a_1^- = 0, \quad b_1^- = 0. \quad (14)$$

Proof of Theorem 2. To prove this, we must demonstrate that any CPWL function f can be expressed as the difference of two convex CPWL functions $f = \max_{j \in \{1, \dots, P^+\}} g_j^+ - \max_{j \in \{1, \dots, P^-\}} g_j^-$, with $g_1^- = 0$. According to Proposition 1, there is a set of affine functions $\{f_1^+, \dots, f_{P^+}^+, f_1^-, \dots, f_{P^-}^-\}$ such that f can be expressed as $f = \max_{j \in \{1, \dots, P^+\}} f_j^+ - \max_{k \in \{1, \dots, P^-\}} f_k^-$. Let $g_j^c = f_j^c - f_1^-$, $\forall j \in \{1, \dots, P^c\}, c \in \{+, -\}$. The function g_j^c is affine because it is the difference of two affine functions. Moreover, $g_1^- = f_1^- - f_1^- = 0$. According to Lemma 3, we have $\max_{j \in \{1, \dots, P^c\}} (g_j^c) = \max_{j \in \{1, \dots, P^c\}} (f_j^c) - f_1^-$. Therefore, we can rewrite f as $f = \max_{j \in \{1, \dots, P^+\}} (f_j^+) - \max_{k \in \{1, \dots, P^-\}} (f_k^-) + f_1^- - f_1^- = \max_{j \in \{1, \dots, P^+\}} (g_j^+) - \max_{k \in \{1, \dots, P^-\}} (g_k^-)$. \square

Note that Equation (14) affects not only the set of possible values for a_1^-, b_1^- , but also for all a_j^c, b_j^c . Also note that the fixed values of zero for a_1^-, b_1^- are chosen arbitrarily; any value could be imposed and serve as a valid tightening constraint for the problem. Imposing Equation (14) effectively reduces the dimensionality of the search space by $(d + 1)$. Formally, Equation (14) does not introduce any additional linear constraints; instead, it introduces tight bounds on $(d + 1)$ linear variables.

5.2. Sorting the Affine Pieces

Let $a_j^c = \begin{pmatrix} a_{j,1}^c \\ \vdots \\ a_{j,d}^c \end{pmatrix} \in \mathbb{R}^d$ denote the linear coefficients of the affine piece f_j^c .

Theorem 3. *The following equation constitutes a valid tightening constraint of MILP1(Q):*

$$a_{j,1}^c \leq a_{j+1,1}^c, \quad j \in \{1, \dots, P^c - 1\}, \quad c \in \{+, -\}. \quad (15)$$

Proof of Theorem 3. The problem MILP1(Q) exhibits symmetries with respect to the group of variables $(a_j^+, b_j^+)_{j=1, \dots, P^+}$ and $(a_k^-, b_k^-)_{j=1, \dots, P^-}$, respectively. This means that permuting the affine pieces of f^+ , or f^- , does not affect the solution of the problem. Therefore, we can arbitrarily sort the affine pieces of f^+ and f^- based on the ascending order of the values $(a_{j,1}^+)_{j=1, \dots, P^+}$ and $(a_{k,1}^-)_{k=1, \dots, P^-}$, respectively. \square

It is important to note that, after sorting the affine pieces of f^+ and f^- , the procedure outlined in the proof of Theorem 2 can be applied to impose Equation (14) with no impact on the CPWL solution. Thus, Equations (14) and (15) are compatible and can be applied simultaneously with no impact on the CPWL solution. The number of possible arrangements for P^c affine pieces is $P^c!$, which implies that implementing Equation (15) reduces the feasible region by a factor $P^+!P^-!$ of its original extent. Additionally, applying Equation (15) introduces $P^+ + P^- - 2$ constraints to the MILP problem.

5.3. Imposing $d+1$ Points per Affine Domain of f^c

Theorem 4. *Adding the following equation to MILP1(Q) reduces the feasible region to a superset of $CPWL^*(S, \varepsilon, P^+, P^-)$. In other words, some non-well-behaved solutions may be eliminated from the new feasible region, but all well-behaved solutions are preserved.*

$$\sum_{i=1}^N \delta_{i,j}^c \geq d + 1, \quad j \in \{1, \dots, P^c\}, \quad c \in \{+, -\}. \quad (16)$$

Proof of Theorem 4. Let $f = f^+ - f^-$ be a well-behaved CPWL solution. Let D_j^+ be one of the affine domains of f^+ . According to Proposition 1, $Int(D_j^+) \neq \emptyset$ and $D = \cup_{k=1}^{P^-} D_k^-$. Therefore, there exists a k such that $Int(D_j^+ \cap D_k^-) \neq \emptyset$. According to Proposition 2, $D_p = D_j^+ \cap D_k^-$ is an affine domain of f . According to Definition 6, $|D_p \cap S_{\mathbb{R}^d}| \geq d + 1$. Furthermore, $D_p \subset D_j^+$ implies $D_p \cap S_{\mathbb{R}^d} \subset D_j^+ \cap S_{\mathbb{R}^d}$. Therefore, we have $|D_j^+ \cap S_{\mathbb{R}^d}| \geq |D_p \cap S_{\mathbb{R}^d}| \geq d + 1$. Similarly, for an affine domain D_k^- of f^- , we obtain $|D_k^- \cap S_{\mathbb{R}^d}| \geq d + 1$. In other words, each affine domain of f^+ and f^- contains at least $d + 1$ points. \square

Note that the inequality from Equation (5), which allows each point to belong to more than one affine domain of f^+ or f^- , is critical for utilizing Equation (16). If we were instead imposing a single affine domain per point, as in the formulation from Kazda and Li (2021), Equation (16) would become too restrictive. In fact, the combined requirement of having only one affine domain per point and at least $(d + 1)$ points per affine domain would imply that $N \geq \max(P^+, P^-)(d + 1)$, making the MILP problem infeasible if that condition was not satisfied. Using Equation (16) eliminates some non-well-behaved CPWL solutions from the feasible region. However, all well-behaved solutions are preserved. Owing to Theorem 1, it is proven that this does not impact the quality of the feasible CPWL solutions, as any CPWL ε -approximation of S has a well-behaved version. Implementing Equation (16) adds $P^+ + P^-$ constraints to the MILP problem.

5.4. Imposing $d+1$ Points per Affine Domain of f

Alternatively, we can impose a set of constraints that further reduces the feasible region to the set of well-behaved CPWL solutions. This approach requires the introduction of two additional sets of variables, $\beta_{i,j,k}$ and $\gamma_{j,k}$. In the following equations, the indices $i \in \{1, \dots, N\}$, $j \in \{1, \dots, P^+\}$, $k \in \{1, \dots, P^-\}$, unless otherwise specified.

Theorem 5. Adding the following equations to MILP1(Q) reduces the feasible region to $CPWL^*(S, \varepsilon, P^+, P^-)$:

$$\beta_{i,j,k} \leq \delta_{i,j}^+ \quad (17)$$

$$\beta_{i,j,k} \leq \delta_{i,k}^- \quad (18)$$

$$\beta_{i,j,k} \geq \delta_{i,j}^+ + \delta_{i,k}^- - 1, \quad (19)$$

$$\beta_{i,j,k} \leq \gamma_{j,k} \quad (20)$$

$$\sum_{i=1}^N \beta_{i,j,k} \geq (d+1)\gamma_{j,k} \quad (21)$$

$$0 \leq \beta_{i,j,k} \leq 1, \quad (22)$$

$$0 \leq \gamma_{j,k} \leq 1. \quad (23)$$

Proof of Theorem 5. Equations (17)–(19) and (22) define $\beta_{i,j,k}$ as an indicator variable which is equal to one if $x_i \in D_j^+ \cap D_k^-$ and zero otherwise—that is, $\beta_{i,j,k} = \delta_{i,j}^+ \wedge \delta_{i,k}^-$. Equations (20) and (23) define $\gamma_{j,k}$ as an indicator variable which is equal to one if $D_j^+ \cap D_k^-$ contains at least one point—that is, $\bigvee_{i=1}^N \beta_{i,j,k} = 1 \Rightarrow \gamma_{j,k} = 1$. Finally, Equation (21) formulates the condition that if one point is in $D_j^+ \cap D_k^-$, then $D_j^+ \cap D_k^-$ must contain at least $d+1$ points—that is, $\gamma_{j,k} = 1 \Rightarrow \sum_{i=1}^N \beta_{i,j,k} \geq (d+1)$. Formally, $\beta_{i,j,k}$ and $\gamma_{j,k}$ should be defined as binary variables. However, the tight constraints from Equations (17)–(21) involving the binary variables $\delta_{i,j}^c$ ensure that $\beta_{i,j,k}$ and $\gamma_{j,k}$ can only take values in $\{0, 1\}$. \square

Equations (17)–(23) involve the use of $P^+P^-(N+1)$ additional variables, $P^+P^-(N+1)$ variable bounds, and $P^+P^-(4N+1)$ additional constraints. These constraints restrict the feasible space to be $CPWL^*(S, \varepsilon, P^+, P^-)$. Note that the indicator variable $\gamma_{j,k}$ is necessary in Equation (21) because the intersection of an affine domain of f^+ and f^- may be empty.

Remark 5. Equation (16) is implied by Equations (17)–(23), meaning that the latter dominate the former. In other words, imposing $d+1$ points per affine domain of f implies, and is more restrictive than, imposing $d+1$ points per affine domain of f^c .

5.5. Tightening the Big-M Parameters

In order to identify tight big-M values, we first need to define several sets. Let A denote the set of all affine functions $D \rightarrow \mathbb{R}$. Let $[S]^{d+1}$ denote the set of all subsets of S composed of $d+1$ points. Let $s = (x_{i_k}, z_{i_k})_{k=1, \dots, d+1} \in [S]^{d+1}$. We define the set $B(s, \varepsilon) = \{(x_{i_k}, z_{i_k} + \ell_k)_{k=1, \dots, d+1} : (\ell_k)_{k=1, \dots, d+1} \in \{-\varepsilon, +\varepsilon\}^{d+1}\}$. We define the set $A_{\varepsilon, s} = \{g \in A : g \varepsilon\text{-approximates } s\}$ and $A_{\varepsilon, s}^* = \{g \in A : g \text{ interpolates } s' \in B(s, \varepsilon)\}$. We also note $A_\varepsilon(S) = \bigcup_{s \in [S]^{d+1}} A_{\varepsilon, s}$ and $A_\varepsilon^*(S) = \bigcup_{s \in [S]^{d+1}} A_{\varepsilon, s}^*$. For convenience, let $-c = \begin{cases} "-" & \text{if } c = "+" \\ "+" & \text{if } c = "-" \end{cases}$. Finally, given a CPWL solution $f = f^+ - f^-$, we define $f_{j,k} = f_j^+ - f_k^-$.

Proposition 5. The set $A_\varepsilon^*(S)$ contains at most $\binom{N}{d+1} 2^{d+1}$ affine functions.

Proof of Proposition 5. First, we have $|[S]^{d+1}| = \binom{N}{d+1}$. Additionally, $|A_{\varepsilon, s}^*| = |B(s, \varepsilon)| = | \{-\varepsilon, +\varepsilon\}^{d+1} | = 2^{d+1}$.

Therefore, $|A_\varepsilon^*(S)| = |\bigcup_{s \in [S]^{d+1}} A_{\varepsilon, s}^*| \leq \sum_{s \in [S]^{d+1}} |A_{\varepsilon, s}^*| = \binom{N}{d+1} 2^{d+1}$. \square

Lemma 4. Let $f \in CPWL^*(S, \varepsilon, P^+, P^-)$, $x \in D$. Let $(j, k) \in \{1, \dots, P^+\} \times \{1, \dots, P^-\} : \text{Int}(D_j^+ \cap D_k^-) \neq \emptyset$. Then, we have:

$$\min_{g \in A_\varepsilon^*(S)} g(x) \leq f_{j,k}(x) \leq \max_{g \in A_\varepsilon^*(S)} g(x).$$

Proof of Lemma 4. Because f is well-behaved, $f_{j,k}$ is an affine piece of f that ε -approximates a subset $s \in [S]^{d+1}$. Therefore, $f_{j,k} \in A_{\varepsilon, s} \subset A_\varepsilon(S)$. As a result, $\min_{g \in A_\varepsilon(S)} g(x) \leq f_{j,k}(x) \leq \max_{g \in A_\varepsilon(S)} g(x)$. What remains to be proven is that the extrema of $g(x)$ on $A_\varepsilon(S)$ can be found on $A_\varepsilon^*(S)$. Note that $\max_{g \in A_{\varepsilon, s}} g(x)$ is equal to the optimal objective value of the LP optimization problem $LP_{\max}(s, x, \varepsilon)$:

$$\max g(x), \quad \text{s.t. } g \in A, \quad -\varepsilon \leq g(x_{i_k}) - z_{i_k} \leq \varepsilon, \quad (x_{i_k}, z_{i_k}) \in s.$$

Similarly, $\min_{g \in A_{\varepsilon, s}} g(\mathbf{x})$ is equal to the optimal objective value of $LP_{\min}(s, \mathbf{x}, \varepsilon)$. In addition, $A_{\varepsilon, s}$ corresponds to the feasible region of $LP_{\min}(s, \mathbf{x}, \varepsilon)$ and $LP_{\max}(s, \mathbf{x}, \varepsilon)$, whereas $A_{\varepsilon, s}^*$ corresponds to the vertices—that is, extreme points—of $A_{\varepsilon, s}$, where the optimal solution is located. Consequently, $\max_{g \in A_{\varepsilon}(S)} g(\mathbf{x}) = \max_{s \in [S]^{d+1}} (\max_{g \in A_{\varepsilon, s}}(g(\mathbf{x}))) = \max_{s \in [S]^{d+1}} (\max_{g \in A_{\varepsilon, s}^*}(g(\mathbf{x}))) = \max_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x})$. Similarly, $\min_{g \in A_{\varepsilon}(S)} g(\mathbf{x}) = \min_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x})$. \square

Lemma 5. Let $f \in CPWL^*(S, \varepsilon, P^+, P^-)$, $\mathbf{x} \in D$, $c \in \{+, -\}$, $(j, k) \in \{1, \dots, P^c\}^2$. Then,

$$|f_j^c(\mathbf{x}) - f_k^c(\mathbf{x})| \leq \min(P^c - 1, P^{-c}) \left(\max_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x}) - \min_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x}) \right).$$

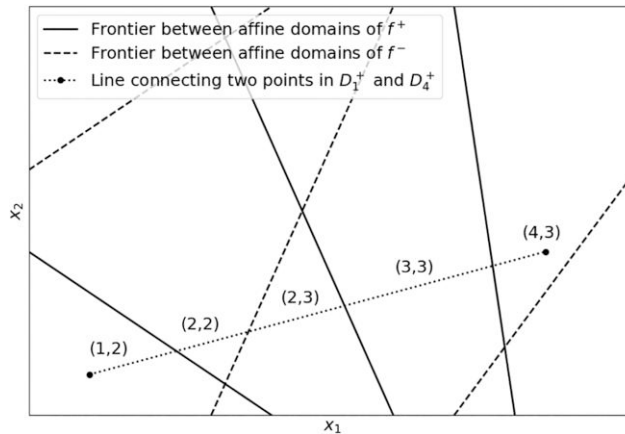
Proof of Lemma 5. We prove the case $c = +$, the case $c = -$ being symmetric. Let $\mathbf{x}_j \in D_j^+$, $\mathbf{x}_k \in D_k^+$. By convexity of D , there exists a line segment L connecting \mathbf{x}_j to \mathbf{x}_k that is contained in D . L crosses T affine domains of f , each of which is the intersection of an affine domain of f^+ and f^- . Let $(j_m, k_m)_{m=1, \dots, T}$ be such that L crosses the nonempty domains $D_{j_m}^+ \cap D_{k_m}^-$ in ascending order of m when traversing from \mathbf{x}_j to \mathbf{x}_k . An example is depicted in Figure 3 for $d = 2$. We have $j_1 = j$ and $j_T = k$. Without loss of generality, we assume that at each step m , the line segment L transitions to either a new affine domain of f^+ or a new affine domain of f^- , but not both. In other words, $(j_m \neq j_{m+1}) \oplus (k_m \neq k_{m+1})$, $\forall m = 1, \dots, T - 1$. Because the intersection of L with an affine domain is convex, $\{m : j_m = j\}$ and $\{m : k_m = k\}$ are sets of consecutive numbers—that is, without gaps. Consequently, j_m and k_m can change value at most $P^+ - 1$ and $P^- - 1$ times, respectively. That is, $|\{m : j_m \neq j_{m+1}\}| \leq P^+ - 1$. Thus, we have:

$$\begin{aligned} f_j^+(\mathbf{x}) - f_k^+(\mathbf{x}) &= f_{j_1}^+(\mathbf{x}) - f_{j_T}^+(\mathbf{x}) \stackrel{(a)}{=} \sum_{m=1}^{T-1} (f_{j_m}^+(\mathbf{x}) - f_{j_{m+1}}^+(\mathbf{x})) \stackrel{(b)}{=} \sum_{m: j_m \neq j_{m+1}} (f_{j_m}^+(\mathbf{x}) - f_{j_{m+1}}^+(\mathbf{x})) \\ &\stackrel{(c)}{=} \sum_{m: j_m \neq j_{m+1}} (f_{j_m}^+(\mathbf{x}) - f_{k_m}^-(\mathbf{x}) + f_{k_m}^-(\mathbf{x}) + f_{j_{m+1}}^+(\mathbf{x})) = \sum_{m: j_m \neq j_{m+1}} (f_{j_m, k_m}(\mathbf{x}) - f_{j_{m+1}, k_{m+1}}(\mathbf{x})) \\ &\stackrel{(d)}{\leq} \sum_{m: j_m \neq j_{m+1}} \left(\max_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x}) - \min_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x}) \right) \stackrel{(e)}{\leq} (P^+ - 1) \left(\max_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x}) - \min_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x}) \right). \end{aligned}$$

(a) uses a telescoping sum, (b) is due to the fact that $f_{j_m}^+(\mathbf{x}) = f_{j_{m+1}}^+(\mathbf{x})$ when $j_m = j_{m+1}$, (c) derives from $(j_m \neq j_{m+1}) \Rightarrow (k_m = k_{m+1})$, (d) is a result of Lemma 4, and (e) stems from $|\{m : j_m \neq j_{m+1}\}| \leq P^+ - 1$. We note K the set of unique values of $\{k_m, m = 1, \dots, M\}$, $\underline{m}(p) = \min_{k_m=p}(m)$, and $\overline{m}(p) = \max_{k_m=p}(m)$. We have:

$$\begin{aligned} f_j^+(\mathbf{x}) - f_k^+(\mathbf{x}) &= \sum_{p \in K} \left(\sum_{\substack{m: j_m \neq j_{m+1}, \\ k_m = k_{m+1} = p}} (f_{j_m, k_m}(\mathbf{x}) - f_{j_{m+1}, k_{m+1}}(\mathbf{x})) \right) \\ &\stackrel{(f)}{=} \sum_{p \in K} (f_{\underline{m}(p), k_{\underline{m}(p)}}(\mathbf{x}) - f_{\overline{m}(p), k_{\overline{m}(p)}}(\mathbf{x})) \stackrel{(g)}{\leq} P^- \left(\max_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x}) - \min_{g \in A_{\varepsilon}^*(S)} g(\mathbf{x}) \right). \end{aligned}$$

Figure 3. Example of a Line Connecting Two Points from Distinct Affine Domains of f^+ for $d = 2$



Notes. Here, both f^+ and f^- have four affine domains, and the line connects a point in D_1^+ to a point in D_4^+ . The line crosses five pairs of affine domains in f^+ and f^- : $((1, 2), (2, 2))$, $((2, 2), (2, 3))$, $((2, 3), (3, 3))$, $((3, 3), (4, 3))$. We have $f_1^+(\mathbf{x}) - f_4^+(\mathbf{x}) = (f_{1,2}(\mathbf{x}) - f_{2,2}(\mathbf{x})) + (f_{2,3}(\mathbf{x}) - f_{4,3}(\mathbf{x}))$.

(f) uses a telescoping sum on $\{\underline{m}(p), \dots, \overline{m}(p)\}$, and (g) is due to Lemma 4 and the fact that $|K| \leq P^-$. As a result,

$$f_j^+(\mathbf{x}) - f_k^+(\mathbf{x}) \leq \min(P^+ - 1, P^-) \left(\max_{g \in A_\varepsilon^+(S)} g(\mathbf{x}) - \min_{g \in A_\varepsilon^-(S)} g(\mathbf{x}) \right).$$

Similarly, the lower bound of $f_j^+(\mathbf{x}) - f_k^+(\mathbf{x})$ is proven by flipping the inequality in in (d) and (g). \square

Theorem 6. Assuming we only consider well-behaved CPWL solutions, the following value is a tight value for the parameter M_n^c in Equation (4):

$$M_n^c = \min(P^c - 1, P^{-c}) \left(\max_{g \in A_\varepsilon^+(S)} g(\mathbf{x}_n) - \min_{g \in A_\varepsilon^-(S)} g(\mathbf{x}_n) \right). \quad (24)$$

Proof of Theorem 6. Let $n \in \{1, \dots, N\}$, $c \in \{+, -\}$, $j \in \{1, \dots, P^c\}$. Let k be such that $\mathbf{x}_n \in D_k^c$ —that is, $f^c(\mathbf{x}_n) = f_k^c(\mathbf{x}_n)$. Owing to Lemma 5, we have:

$$f^c(\mathbf{x}_n) - f_j^c(\mathbf{x}_n) = f_k^c(\mathbf{x}_n) - f_j^c(\mathbf{x}_n) \leq \min(P^c - 1, P^{-c}) \left(\max_{g \in A_\varepsilon^+(S)} g(\mathbf{x}_n) - \min_{g \in A_\varepsilon^-(S)} g(\mathbf{x}_n) \right). \quad \square$$

Herein, it is assumed that the value of the parameter M_n^c from Equation (4) is the one described in Theorem 6. Note that the big-M values defined in Equation (24) are only valid when considering the solution space of well-behaved CPWL solutions. This implies that using these values may eliminate some CPWL solutions that are not well-behaved from the feasible region, although not necessarily all of them.

5.6. Bounding the Variables

We define the following parameters:

$$\underline{a}_r = \min_{g \in A_\varepsilon^+(S)} \frac{\partial g}{\partial x_r}, \quad \overline{a}_r = \max_{g \in A_\varepsilon^-(S)} \frac{\partial g}{\partial x_r}, \quad (25)$$

$$\underline{b} = \min_{g \in A_\varepsilon^+(S)} g(\mathbf{0}), \quad \overline{b} = \max_{g \in A_\varepsilon^-(S)} g(\mathbf{0}), \quad (26)$$

$$\overline{a}_r' = \min(P^- - 1, P^+) (\overline{a}_r - \underline{a}_r), \quad (27)$$

$$\overline{b}' = \min(P^- - 1, P^+) (\overline{b} - \underline{b}). \quad (28)$$

Lemma 6. Let $f \in \text{CPWL}^*(S, \varepsilon, P^+, P^-)$. Let $(j, k) \in \{1, \dots, P^+\} \times \{1, \dots, P^-\} : \text{Int}(D_j^+ \cap D_k^-) \neq \emptyset$. Then, we have:

$$\begin{aligned} \underline{a}_r &\leq a_{j,r}^+ - a_{k,r}^- \leq \overline{a}_r, \quad r \in \{1, \dots, d\} \\ \underline{b} &\leq b_j^+ - b_k^- \leq \overline{b}. \end{aligned}$$

Proof of Lemma 6. The partial derivatives of an affine function are equal to its linear coefficients, and its value in $\mathbf{0}$ corresponds to its bias term. In particular, $\frac{\partial f_{j,k}}{\partial x_r} = a_{j,r}^+ - a_{k,r}^-$ and $f_{j,k}(\mathbf{0}) = b_j^+ - b_k^-$. Because $f_{j,k} \in A_\varepsilon(S)$, we have $\min_{g \in A_\varepsilon(S)} \frac{\partial g}{\partial x_r} \leq \frac{\partial f_{j,k}}{\partial x_r} \leq \max_{g \in A_\varepsilon(S)} \frac{\partial g}{\partial x_r}$ and $\min_{g \in A_\varepsilon(S)} g(\mathbf{0}) \leq f_{j,k}(\mathbf{0}) \leq \max_{g \in A_\varepsilon(S)} g(\mathbf{0})$. Similar to Lemma 5, we can show that the extrema of $\frac{\partial g}{\partial x_r}$ and $g(\mathbf{0})$ on $A_\varepsilon(S)$ can be found on $A_\varepsilon^*(S)$. Let $s \in [S]^{d+1}$. $\max_{g \in A_{\varepsilon,s}} \frac{\partial g}{\partial x_r}$, $-\min_{g \in A_{\varepsilon,s}} \frac{\partial g}{\partial x_r}$, $\max_{g \in A_{\varepsilon,s}} g(\mathbf{0})$, $-\min_{g \in A_{\varepsilon,s}} g(\mathbf{0})$ are the optimal objective values of the following LP optimization problem, with $Q = a_r$, $Q = -a_r$, $Q = b$, $Q = -b$, respectively.

$$\max Q, \quad \text{s.t.} \quad -\varepsilon \leq \sum_{r=1}^d a_r x_{i_k,r} + b - z_{i_k} \leq \varepsilon, \quad (\mathbf{x}_{i_k}, z_{i_k}) \in s.$$

Therefore, the extrema of $\frac{\partial g}{\partial x_r}$ and $g(\mathbf{0})$ on $A_{\varepsilon,s}$ can be found on the vertices $A_{\varepsilon,s}^*$. It follows that the extrema on $A_\varepsilon(S)$ can be found on $A_\varepsilon^*(S)$. \square

Lemma 7. Let $f \in \text{CPWL}^*(S, \varepsilon, P^+, P^-)$, $(j, k) \in \{1, \dots, P^-\}^2$. Then,

$$\begin{aligned} |a_{j,r}^- - a_{k,r}^-| &\leq \bar{a}'_r, \quad \forall r \in \{1, \dots, d\} \\ |b_j^- - b_k^-| &\leq \bar{b}' \end{aligned}$$

Proof of Lemma 7. This is demonstrated by using the same sum decomposition as in Lemma 5 but for $a_{j,r}^-$, b_j^- instead of $f_j^+(x)$. The result follows from applying Lemma 6 to identify a similar inequality as (d) and (g). \square

Theorem 7. For well-behaved CPWL solutions, the following equations represent valid upper and lower bounds of the linear variables:

$$z_i - \varepsilon \leq f(x_i) \leq z_i + \varepsilon, \quad i \in \{1, \dots, N\}, \quad (29)$$

$$0 \leq f^-(x_i) \leq M_i^-, \quad i \in \{1, \dots, N\}, \quad (30)$$

$$z_i - \varepsilon \leq f^+(x_i) \leq z_i + \varepsilon + M_i^-, \quad i \in \{1, \dots, N\}, \quad (31)$$

$$-\bar{a}'_r \leq a_{j,r}^- \leq \bar{a}'_r, \quad j \in \{1, \dots, P^-\}, r \in \{2, \dots, d\}, \quad (32)$$

$$0 \leq a_{j,1}^- \leq \bar{a}'_1, \quad j \in \{1, \dots, P^-\}, \quad (33)$$

$$\underline{a}_r - \bar{a}'_r \leq a_{j,r}^+ \leq \bar{a}_r + \bar{a}'_r, \quad j \in \{1, \dots, P^+\}, r \in \{2, \dots, d\}, \quad (34)$$

$$\underline{a}_1 \leq a_{j,1}^+ \leq \bar{a}_1 + \bar{a}'_1, \quad j \in \{1, \dots, P^+\}, \quad (35)$$

$$-\bar{b}' \leq b_j^- \leq \bar{b}', \quad j \in \{1, \dots, P^-\}, \quad (36)$$

$$\underline{b} - \bar{b}' \leq b_j^+ \leq \bar{b} + \bar{b}', \quad j \in \{1, \dots, P^+\}. \quad (37)$$

Proof of Theorem 7. The bounds from Equation (29) derive from Equations (6) and (7). Equation (30) is based on Equation (14), using $f^-(x_i) = \max(f_1^-(x_i), f_2^-(x_i), \dots, f_{P^-}^-(x_i)) = \max(0, f_2^-(x_i), \dots, f_{P^-}^-(x_i)) \geq 0$. In addition, $\exists k \in \{1, \dots, P^-\} : x_i \in D_k^-$. Thus, according to Lemma 5, $f^-(x_i) = f_k^-(x_i) = f_k^-(x_i) - f_1^-(x_i) \leq M_i^-$. Equation (31) derives from Equations (29) and (30) combined with $f^+(x_i) = f(x_i) + f^-(x_i)$. Using Equation (14) and Lemma 7, $|a_{j,r}^-| = |a_{j,r}^- - a_{1,r}^-| \leq \bar{a}'_r$, leading to Equation (32). Equation (33) arises from Equation (15): $a_{j,1}^- \geq \dots \geq a_{1,1}^- \geq 0$. Next, combining Equation (32), Lemma 6, and k such that $\text{Int}(D_j^+ \cap D_k^-) \neq \emptyset$, we deduce: $a_{j,r}^+ = a_{j,r}^+ - a_{k,r}^- + a_{k,r}^-$ and $\underline{a}_r - \bar{a}'_r \leq a_{j,r}^+ - a_{k,r}^- + a_{k,r}^- \leq \bar{a}_r + \bar{a}'_r$, leading to Equation (34). Equation (35) stems from Equations (33) and (34). Further, applying Equation (14) and Lemma 7, we deduce $|b_j^-| = |b_j^- - b_1^-| \leq \bar{b}'$, resulting in Equation (36). Finally, Equation (37) stems from Equation (36), Lemma 6, and k such that $\text{Int}(D_j^+ \cap D_k^-) \neq \emptyset$. \square

Note that the bounding values from Equations (30)–(37) are only valid when considering the solution space of well-behaved CPWL solutions. This implies that using these bounding values may eliminate some CPWL solutions that are not well-behaved from the feasible region, although not necessarily all of them. In addition, Equations (32)–(37) apply owing to Equations (14) and (15). This implies that using these values may eliminate some DC representations of the CPWL solutions but not the CPWL solutions themselves.

Proposition 6. The time complexity for calculating the bounds and big-M parameters is $O(2^{d+1}N^{d+2}d)$.

The proof of Proposition 6 is based upon the number of elements in $A_\varepsilon^*(S)$ and the time complexity of inverting a matrix in $M_{d+1}(\mathbb{R})$. A more detailed proof is provided in Section EC.4 of the electronic companion.

Table 1 gives a summary of the tightening strategies, together with their impact on the feasible region and the preprocessing involved.

Remark 6. Note that applying all equations presented in Section 5 is a valid tightening of the MILP problem to the feasible region of the well-behaved CPWL solutions. As a result, applying any subset of equations from Section 5 is also a valid tightening. For example, although the variable bounds from Section 5.6 derive from fixing $f_1^- = 0$ and sorting the affine pieces of f^+ and f^- , applying the variable bounds without sorting or fixing any affine piece is still a valid tightening of the MILP problem. As a matter of fact, any combination of the tightening strategies described here is a valid tightening. This fact is leveraged in Section 6 to identify effective combinations of tightening strategies.

Table 1. Summary of Tightening Strategies

Eq.	Description	# of new constraints or variable bounds ^a	# of new variables	Impact on search region	Preprocessing	Time complexity
(14)	Fix one affine piece	0 ^a $d + 1$	0	Eliminate $d + 1$ dimensions	None	
(15)	Sort the affine pieces	$P^+ + P^- - 2$	0	Reduce the feasible region by a factor of $P^+P^-!$	None	
(16) ^b	Impose $d + 1$ points per affine piece of f^c	$P^+ + P^-$	0	Eliminate some non-well-behaved CPWL solutions	None	
(17)–(23) ^b	Impose $d + 1$ points per affine piece of f	$P^+P^-(4N + 1)$ ^a $P^+P^-(N + 1)$	$P^+P^-(N + 1)$	Eliminate all non-well-behaved CPWL solutions	None	
(24)	Tighten the big-M parameters	0	0	Eliminate some non-well-behaved CPWL solutions	Compute all $g \in A_\varepsilon^*(S)$, evaluate g in N points, compute extrema for each point	$O(2^{d+1}N^{d+2}d)$
(29)–(37)	Bound the variables	0 ^a $3N + (P^+ + P^-)(d + 1)$	0	Bound the search space in all variable dimensions	For all $g \in A_\varepsilon^*(S)$, evaluate the extrema of the coefficients of g	$O(2^{d+1}N^{d+1}d)$

^aIn the number of additional constraints, indicates the number of additional variable bounds.

^bThe two sets of equations are alternative to one another, the first set being dominated by the second set but involving less constraints and variables than the second set.

6. Computational Experiments

This section illustrates the computational performances of the tightening strategies. In Section 6.1, the data sets used to assess the efficiency of the tightening strategies are described. In Section 6.2, different combinations of the tightening strategies are evaluated to identify the ones that have the most impact on the solution time. In addition, the computational performance of the tightening procedure is assessed.

The algorithms are implemented in Python (3.12.7), and the MILP problems are solved with the Gurobi solver (12.0.0). The code used for the computational experiments can be found at Ploussard (2025). Models are run on an Intel 2.3-GHz machine with 16 cores and 32 GB of RAM. For all MILP problems, the relative optimality gap (MIPGap) is set to 10^{-6} . In addition, the primal feasibility tolerance (FeasibilityTol) and integer feasibility tolerance (IntFeasTol) are both set to their minimum value of 10^{-9} to mitigate numerical errors.

The objective function used for all MILP problems is the maximum error—that is, Equation (13).

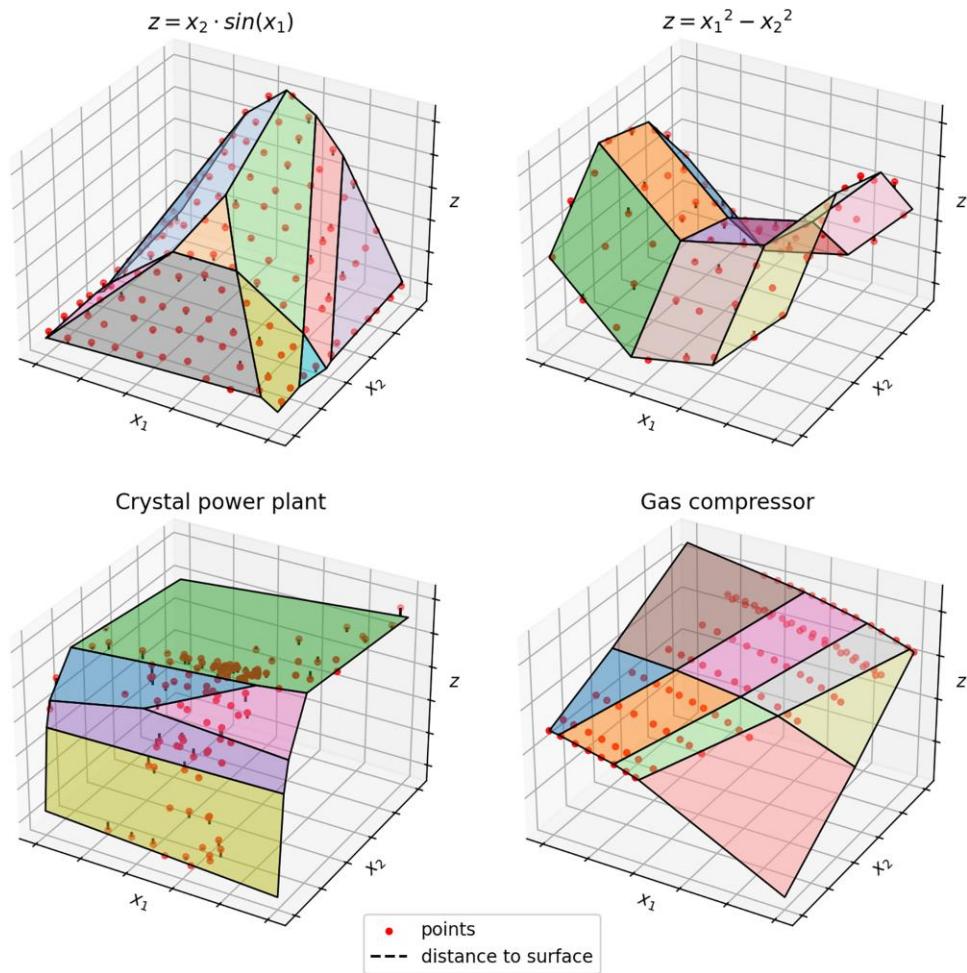
6.1. Description of the Data Sets

To assess the efficiency of the proposed method, six data sets are used. The data sets are summarized in Table 2. Four of the data sets are derived from mathematical functions, and two data sets are based on real-world data. Data sets based on mathematical functions are generated by randomly selecting a certain number of x points in a given domain of \mathbb{R}^d . The first four data sets from Table 2 are two-dimensional and are illustrated by the dots in Figure 4. The last two data sets are three-dimensional (Figure 5). The third data set represents the water-to-power conversion factor (z) at the Crystal hydropower plant based on the forebay elevation (x_1) and the average daily water release (x_2) from the years 2014–2024 (Bureau of Reclamation 2024). The fourth data set represents the discharge temperature of a gas compressor (z) based on the volumetric flow rate (x_1) and rotation speed (x_2) (Marfatia and Li 2022). Data sets were rescaled in all dimensions to prevent numerical issues during the MILP solving process.

Table 2. Data Sets Used for Numerical Experiments

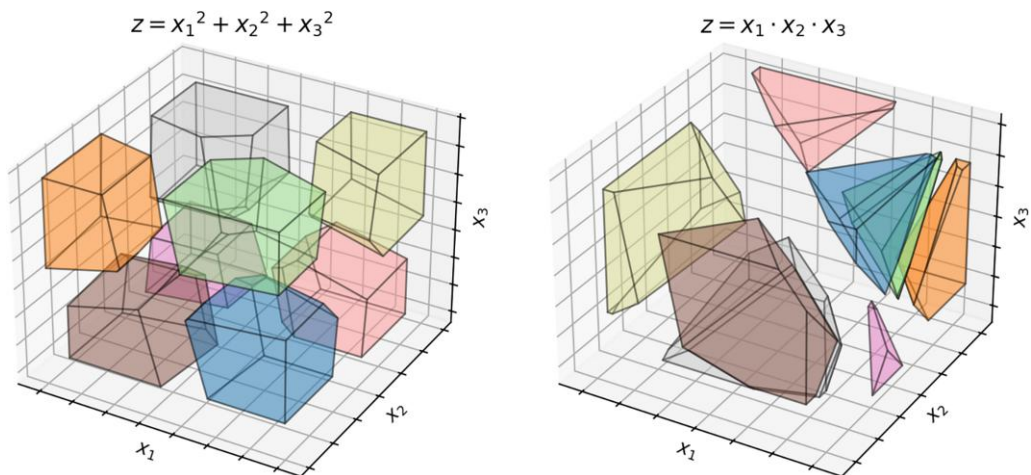
Data set	Number of data points	Dimension d
$z = x_2 \cdot \sin(x_1), (x_1, x_2) \in [0, \pi] \times [0, 1]$	121	2
$z = x_1^2 - x_2^2, (x_1, x_2) \in [-1, 1]^2$	64	2
Crystal power plant, Bureau of Reclamation (2024)	116	2
Gas compressor, Marfatia and Li (2022)	102	2
$z = x_1^2 + x_2^2 + x_3^2, (x_1, x_2, x_3) \in [0, 1]^3$	64	3
$z = x_1 \cdot x_2 \cdot x_3, (x_1, x_2, x_3) \in [0, 1]^3$	64	3

Figure 4. (Color online) Optimal CPWL Fitting for the Two-Dimensional Data Sets



Note. Each polygonal face represents an affine piece of the CPWL function f .

Figure 5. (Color online) Exploded View of the Affine Domains for the Three-Dimensional Data Sets



Note. Each polyhedron represents the domain of an affine piece of f .

Table 3. Combinations of Tightening Strategies Assessed

Combination	Fix one affine piece	Sort the affine pieces	Impose $d + 1$ points per affine piece	Big-M parameters	Bounded variables
C1				Indicator	
C2				Default	
C3				Tight	
C4	✓			Tight	
C5	✓	✓		Tight	
C6	✓	✓		Tight	✓
C7	✓	✓	of f^+ and f^-	Tight	✓
C8	✓	✓	of f	Tight	✓
C9			of f^+ and f^-	Tight	✓
C10			of f	Tight	✓
C11	✓		of f^+ and f^-	Tight	✓

Note. A blank cell indicates that the tightening strategy does not apply to the combination.

6.2. Evaluating Different Combinations of Tightening Strategies

There are several ways to apply the tightening strategies to the MILP problem:

- The first affine piece of f^- may or may not be set to zero (Equation (14)).
- The affine pieces of f^+ and f^- may or may not be sorted (Equation (15)).
- We can impose $d + 1$ points either for each affine piece of f^+ and f^- (Equation (16)), for each affine piece of f (Equations (17)–(21)), or for no affine piece.
 - For the big-M parameters, we can either use an indicator constraint (Equation (11)), a reasonably large (“default”) big-M value, or the tight big-M values (Equation (24)).
 - The variables may or may not be bounded (Equations (29)–(37)).

In total, there are up to 72 combinations of tightening strategies. Because of this large number, only 11 combinations were selected. The selected combinations, referred to as “C1” to “C11,” are described in Table 3. The “default” big-M value is set equal to the rounded-up maximum value of M_i^c —for example, 700 if the maximum value is 632.8. For each data set, the 11 combinations of tightening strategies are applied to the MILP problem before solving it, and the solution time is measured for each combination. The same number P^+ and P^- of candidate affine pieces and the same upper bound ϵ for the maximum error are used across all combinations. The time limit to solve the MILP problem is set to 7,200 seconds. Numerical results are summarized in Table 4.

For each data set, the optimal objective value of the MILP problem is identical across all combinations. However, solution times noticeably vary across combinations. Results show that there is a clear benefit of using tight big-M parameter values (C3) as opposed to using a default big-M parameter value (C2) or an indicator constraint (C1). Fixing the first affine piece of f^- (C4) without imposing $d + 1$ points per affine piece does not seem to have a significant additional benefit, except for the case “ $z = x_1^2 + x_2^2 + x_3^2$.” Although it reduces the feasible region, sorting the affine pieces of f^- and f^+ has a negative impact on the solution time (C5–C8). This might be due to the fact that these additional constraints increase the solution time of the relaxed LP problem in each node of the MILP search tree without improving the tree search itself—that is, without constraining the binary variables. Because of its negative impact, the strategy of sorting the affine pieces is discarded from the rest of the combinations. Imposing $d + 1$ points per affine piece of f^+ and f^- and bounding the variables (C9 and C11) has a clear

Table 4. Summary of Computation Times for Each Data Set and Combination

Data set	P^+, P^-	ϵ	Preprocessing	Computation time (s)												
				C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11		
$z = x_2 \cdot \sin(x_1)$	2, 6	0.2	9	7,200 ^a	7,200 ^a	7,200 ^a	7,200 ^a	7,200 ^a	7,200 ^a	7,200 ^a	7,200 ^a	7,200 ^a	7,200 ^a	1,877	7,200 ^a	2,554
$z = x_1^2 - x_2^2$	3, 3	0.1	1	7,200 ^a	576.7	112.8	116.9	507.8	382.2	406.4	2,752	98.3	516.6	200.2		
Crystal power plant	1, 5	0.2	8	197.6	8.7	9.9	8.6	72.0	42.0	32.3	31.5	11.4	9.7	20.8		
Gas compressor	2, 4	0.2	5	7,200 ^a	7,200 ^a	1,584	2,090	847.4	1,356	944.9	7,200 ^a	457.2	3,720	303.2		
$z = x_1^2 + x_2^2 + x_3^2$	8, 1	0.1	37	7,200 ^a	3,402	1,441	111.0	7,200 ^a	7,200 ^a	7,200 ^a	7,200 ^a	531.1	478.3	159.5		
$z = x_1 \cdot x_2 \cdot x_3$	2, 4	0.1	38	7,200 ^a	2,740	460.3	558.3	2,496	1,968	974.2	7,200 ^a	698.1	4,390	438.1		

Notes. Summary of computation times for each data set, including the computation time of the preprocessing step and the MILP solving time of each combination of tightening strategies. Bold numbers indicate the two shortest solving times for each data set across all combinations.

^aIndicates that the time limit was reached before meeting the target optimality gap.

positive impact on the solution time. This positive effect can be explained by the reduction in the feasible region of the binary variables $\delta_{i,j}^c$ using few additional constraints ($P^+ + P^-$). Although this feasible region is further reduced when imposing $d + 1$ points per affine piece of f (C10), this constraint has a negative impact on the solution time. This might be due to the larger number of additional variables and constraints required to model the affine pieces of f . The only difference between combinations C9 and C11 is whether the first affine piece of f^- is being fixed. Numerical results show that adding this strategy may either improve or worsen the solution time, depending on the data set—that is, there is not a clear systematic benefit in applying this strategy. When comparing the best solution time of C9 and C11 to the solution time of C2, the reduction factor in solving time ranges from 4 to 23, except for the Crystal power plant case. For the worst solution time, the reduction factor ranges from 3 to 16. Overall, the following combination consistently performs among the best ones: imposing $d + 1$ points for each affine piece of f^+ and f^- (Equation (16)), tightening the big-M parameters (Equation (24)), fixing the first affine piece of f^- (Equation (14)), and bounding all variables (Equations (29)–(37)).

The computation time of the preprocessing step, used to calculate the big-M parameters and variable bounds, is also included in Table 4. The computation time of this step is negligible for two-dimensional data sets. However, the computation time becomes significant for three-dimensional data sets. This is due to the fact that the time complexity of the preprocessing steps increases exponentially with the dimension of the data set (Table 1). Yet, the combined execution time of preprocessing and solving the tightened MILP problem is still shorter than the time required to solve the nontightened MILP problem, demonstrating the efficiency of the tightening approach in reducing overall computational effort.

7. Conclusions

In this paper, we formalize the concept of well-behaved CPWL interpolations, a class of CPWL interpolations in which each affine piece interpolates a number of points greater than the dimension of the interpolated data set. Next, we demonstrate that any CPWL interpolation has a well-behaved version. Then, we introduce several tightening strategies aimed at improving the solution time of the MILP formulation of the CPWL-fitting problem for data sets in general dimensions. Some of the tightening strategies leverage the fact that any CPWL interpolation has a well-behaved version. In addition, we analyze each tightening strategy in terms of additional constraints, additional variables, impact on the feasible region, and time complexity of the preprocessing step. Then, we identify the combinations of tightening strategies that have the most impact on the solution time of the MILP problem. Experimental results show that the tightening procedure significantly reduces the solution time of the MILP problem in most cases. However, the solution time reduction factor can vary significantly, depending on the size and dimension of the data set, with values ranging from 3 to 23 across five of the six case studies. In addition, the computation time of the preprocessing step rapidly increases with the dimension of the data set, making the tightening procedure computationally challenging for high-dimensional data sets ($d \geq 4$). This issue can be partially addressed using parallelization, as the preprocessing step is inherently suited for parallel processing.

The theoretical work presented here applies to data points in general position (no subset of $d + 1$ points is contained within a single hyperplane). There is a need to address the special case where the points are not in general position—for example, if the points are located on a lattice. Research work should also focus on improving the time complexity of the preprocessing algorithms for high dimensions. In addition, future research should aim to conduct a comprehensive comparison between the MILP and NN approaches with respect to computation time, number of affine pieces, and approximation error.

Acknowledgments

This work was authored for the Department of Energy (DOE) Office of Energy Efficiency and Renewable Energy by Argonne National Laboratory, operated by UChicago Argonne LLC under contract number DE-AC02-06CH11357. This study was supported by the HydroWIREs Initiative of DOE's Water Power Technologies Office.

References

- Bärnann A, Burlacu R, Hager L, Kutzner K (2023) An approximation algorithm for optimal piecewise linear interpolations of bounded variable products. *J. Optim. Theory Appl.* 199(2):569–599.
- Bertsimas D, Tsitsiklis J (1997) *Introduction to Linear Optimization* (Athena Scientific, Nashua, NH).
- Boyd S, Vandenberghe L (2004) *Convex Optimization* (Cambridge University Press, Cambridge, UK).
- Bureau of Reclamation (2024) HDB data service. Accessed April 3, 2025, https://www.usbr.gov/lc/region/g4000/riverops/_HdbWebQuery.html.
- Chen KL, Garudadri H, Rao BD (2023) Improved bounds on neural complexity for representing piecewise linear functions. Preprint, submitted January 16, <https://arxiv.org/abs/2210.07236>.

- D'Ambrosio C, Lodi A, Martello S (2010) Piecewise linear approximation of functions of two variables in MILP models. *Oper. Res. Lett.* 38(1):39–46.
- Davis PJ (1975) *Interpolation and Approximation* (Courier Corporation, North Chelmsford, MA).
- DeVore R, Hanin B, Petrova G (2021) Neural network approximation. *Acta Numer.* 30:327–444.
- Duguet A, Ngueveu SU (2022) Piecewise linearization of bivariate nonlinear functions: Minimizing the number of pieces under a bounded approximation error. Ljubić I, Barahona F, Dey SS, Mahjoub AR, eds. *Combinatorial Optimization* (Springer International Publishing, Cham, Switzerland), 117–129.
- Edelsbrunner H, O'Rourke J, Seidel R (1986) Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.* 15(2):341–363.
- Frenzen CL, Sasao T, Butler JT (2010) On the number of segments needed in a piecewise linear approximation. *J. Comput. Appl. Math.* 234(2):437–446.
- Geißler B, Martin A, Morsi A, Schewe L (2012) Using piecewise linear functions for solving MINLPs. Lee J, Leyffer S, eds. *Mixed Integer Non-linear Programming* (Springer, New York), 287–314.
- Gurobi Optimization LLC (2024a) Gurobi Optimizer reference manual. Accessed April 3, 2025, <https://www.gurobi.com>.
- Gurobi Optimization LLC (2024b) Model.addGenConstrIndicator(). Accessed April 3, 2025, https://www.gurobi.com/documentation/current/refman/py_model_agc_indicator.html.
- Huang C (2020) ReLU networks are universal approximators via piecewise linear or constant functions. *Neural Comput.* 32(11):2249–2278.
- Hughes RB, Anderson MR (1996) Simplicity of the cube. *Discrete Math.* 158(1):99–150.
- Kazda K, Li X (2021) Nonconvex multivariate piecewise-linear fitting using the difference-of-convex representation. *Comput. Chemical Engrg.* 150:107310.
- Kazda K, Li X (2024) A linear programming approach to difference-of-convex piecewise linear approximation. *Eur. J. Oper. Res.* 312(2):493–511.
- Kong L, Maravelias CT (2020) On the derivation of continuous piecewise linear approximating functions. *INFORMS J. Comput.* 32(3):531–546.
- Kripfganz A, Schulze R (1987) Piecewise affine functions as a difference of two convex functions. *Optimization* 18(1):23–29.
- Marfatia Z, Li X (2022) Data-driven natural gas compressor models for gas transport network optimization. *Digital Chem. Engrg.* 3:100030.
- Misener R, Floudas CA (2010) Piecewise-linear approximations of multidimensional functions. *J. Optim. Theory Appl.* 145(1):120–147.
- Ploussard Q (2024) Piecewise linear approximation with minimum number of linear segments and minimum error: A fast approach to tighten and warm start the hierarchical mixed integer formulation. *Eur. J. Oper. Res.* 315(1):50–62.
- Ploussard Q (2025) Continuous piecewise linear fitting: Code repository. Accessed April 3, 2025, <https://github.com/quentinplrsrd/cpwl-nd-optimization>.
- Rebennack S, Kallrath J (2015) Continuous piecewise linear delta-approximations for bivariate and multivariate functions. *J. Optim. Theory Appl.* 167(1):102–117.
- Rebennack S, Krasko V (2020) Piecewise linear function fitting via mixed-integer linear programming. *INFORMS J. Comput.* 32(2):507–530.
- Toriello A, Vielma JP (2012) Fitting piecewise linear continuous functions. *Eur. J. Oper. Res.* 219(1):86–95.
- Vielma JP (2015) Mixed integer linear programming formulation techniques. *SIAM Rev.* 57(1):3–57.
- Vielma JP, Ahmed S, Nemhauser G (2010) Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Oper. Res.* 58(2):303–315.
- Warwicker JA, Rebennack S (2022) A comparison of two mixed-integer linear programs for piecewise linear function fitting. *INFORMS J. Comput.* 34(2):1042–1047.
- Warwicker JA, Rebennack S (2023) Generating optimal robust continuous piecewise linear regression with outliers through combinatorial Benders decomposition. *IIEE Trans.* 55(8):755–767.
- Warwicker JA, Rebennack S (2025) Efficient continuous piecewise linear regression for linearising univariate non-linear functions. *IIEE Trans.* 57(3):231–245.