PIECEWISE POLYHEDRAL RELAXATIONS OF MULTILINEAR OPTIMIZATION

JONGEUN KIM*, JEAN-PHILIPPE P. RICHARD†, AND MOHIT TAWARMALANI‡

Abstract. In this paper, we consider piecewise polyhedral relaxations (PPRs) of multilinear optimization problems over axis-parallel hyper-rectangular partitions of their domain. We improve formulations for PPRs by linking components that are commonly modeled independently in the literature. Numerical experiments with ALPINE, an open-source software for global optimization that relies on piecewise approximations of functions, show that the resulting formulations speed-up the solver by an order of magnitude when compared to its default settings. If given the same time, the new formulation can solve more than twice as many instances from our test-set. Most results on piecewise functions in the literature assume that the partition is *regular*. Regular partitions arise when the domain of each individual input variable is divided into nonoverlapping intervals and when the partition of the overall domain is composed of all Cartesian products of these intervals. We provide the first locally ideal formulation for general (non-regular) hyper-rectangular partitions. We also perform experiments that show that, for a variant of tree ensemble optimization problems, a formulation based on non-regular partitions outperforms that over regular ones by an order of magnitude.

Key words. Multilinear optimization, Piecewise modeling, Tree ensembles, Non-regular partitioning

AMS subject classifications. 90C10, 90C23, 90C26

1. Introduction.

We consider multilinear optimization problems of the form

$$
\begin{aligned}
&\text{(1.1a)} \qquad && \max \quad && \boldsymbol{c}_{\boldsymbol{z}}^{\intercal}\boldsymbol{z} + \boldsymbol{c}_{\boldsymbol{w}}^{\intercal}\boldsymbol{w} && \\
&\text{(1.1b)} && \text{s.t.} && A_{\boldsymbol{z}}\boldsymbol{z} + A_{\boldsymbol{w}}\boldsymbol{w} \leq \boldsymbol{b}, && \\
&\text{(1.1c)} && && w_j = f_j(\boldsymbol{z}_{I_j}), && \forall j \in [n_w], \\
&\text{(1.1d)} && && \ell_i \leq z_i \leq u_i, && \forall i \in [n],
\end{aligned}
$$

where n, n_w, $n_A \in \mathbb{Z}_+$, $\boldsymbol{c}_{\boldsymbol{z}} \in \mathbb{R}^n$, $\boldsymbol{c}_{\boldsymbol{w}} \in \mathbb{R}^{n_w}$, $A_{\boldsymbol{z}} \in \mathbb{R}^{n_A \times n}$, $A_{\boldsymbol{w}} \in \mathbb{R}^{n_A \times n_w}$, $\boldsymbol{b} \in \mathbb{R}^{n_A}$, $\boldsymbol{\ell}$ and $\boldsymbol{u} \in \mathbb{R}^n$, $f_j : \mathbb{R}^{|I_j|} \mapsto \mathbb{R}$ is a multilinear function of variables z_k with indices k in $I_j \subseteq [n]$ for all $j \in [n_w]$, $[a] := \{1, \ldots, a\}$ for positive integer a, and where we use bold lowercase letters to denote vectors. Throughout this paper, we use a regular lowercase symbol to represent a variable (such as $\boldsymbol{z}$) and use a bar or a hat above the symbol (such as $\bar{\boldsymbol{z}}$ and $\hat{\boldsymbol{z}}$) to represent the variable at a certain point. Motivated by advances in integer programming solvers, there has been an interest in constructing discrete relaxations for mixed-integer nonlinear programming (MINLP) problems [8, 12]. A strategy adopted by MINLP solvers such as ALPINE [22, 23, 29] and ANTIGONE [20] is to introduce new variables for univariate functions and then use discretization strategies to relax (1.1).

In this paper, we develop insights into this latter relaxation. We refer to subsets of $\mathbb{R}^n$ defined by constraints of the form (1.1d), with $\boldsymbol{\ell} < \boldsymbol{u}$, as *hyper-rectangles*. We denote the hyper-rectangle with lower bounds $\boldsymbol{\ell}$ and upper bounds $\boldsymbol{u}$ as $\mathcal{Z}(\boldsymbol{\ell}, \boldsymbol{u})$, which we abbreviate as $\mathcal{Z}$ when parameters $\boldsymbol{\ell}$ and $\boldsymbol{u}$ are clear from the context. For

*Department of Industrial and Systems Engineering, University of Minnesota-Twin Cities, MN 55455 (kim00623@umn.edu)

†Department of Industrial and Systems Engineering, University of Minnesota-Twin Cities, MN 55455 (jrichar@umn.edu)

‡Krannert School of Management, Purdue University, IN 47907 (mtawarma@purdue.edu)

$I \subseteq [n]$, we denote by $\mathcal{Z}_I$ the hyper-rectangle $\mathcal{Z}(\boldsymbol{\ell}_I, \boldsymbol{u}_I)$ obtained by projecting $\mathcal{Z}$ over the space of $\boldsymbol{z}_I$ variables.

We investigate mixed-integer programming (MIP) models for the type of piecewise polyhedral relaxations of (1.1) over hyper-rectangular partitions that we describe next. A collection $\{Q_i\}_{i\in[L]}$ of full-dimensional subsets of a compact set $S \subseteq \mathbb{R}^n$ is a *partition* of S if (i) $\bigcup_{i\in[L]} Q_i = S$ and (ii) the interiors of $\{Q_i\}_{i\in[L]}$ are pairwise disjoint. A partition $\{Q_i\}_{i\in[L]}$ of S is said to be *polyhedral* if Q_i is a polytope for all $i \in [L]$. We say that a polyhedral partition $\{Q_i\}_{i\in[L]}$ of S is *hyper-rectangular* (or equivalently that $\{Q_i\}_{i\in[L]}$ is a *hyper-rectangular partition (HP)*) if Q_i is a hyper-rectangle for all $i \in [L]$.

Let $h(\boldsymbol{z}) : \mathbb{R}^n \mapsto \mathbb{R}$ be a function whose domain is $\mathcal{Z}(\boldsymbol{\ell}, \boldsymbol{u})$ for some $\boldsymbol{\ell}, \boldsymbol{u} \in \mathbb{R}^n$ and let $\{Q_i\}_{i\in[L]}$ be a polyhedral partition of $\mathcal{Z}$. We denote by $\operatorname{vert} S$ the set of the extreme points of polytope S. A *piecewise polyhedral relaxation (PPR)* of the graph of h over $\{Q_i\}_{i\in[L]}$ is defined as a set $S = \bigcup_{i\in[L]} \bar{Q}_i$ where $\bar{Q}_i$ is a polytope in $\mathbb{R}^{n+1}$, $\operatorname{vert} Q_i = \operatorname{proj}_{\boldsymbol{z}} \operatorname{vert} \bar{Q}_i$, and $\{(\boldsymbol{z}, w) \in Q_i \times \mathbb{R} \mid w = h(\boldsymbol{z})\} \subseteq \bar{Q}_i$ for all $i \in [L]$. We remark that when h is a piecewise function defined over a partition $\{Q_i\}_{i\in[L]}$ of $\mathcal{Z}$, PPR can be naturally used to build a relaxation using the same partition. In general, there may not exist a PPR that is contained in all PPRs that can be constructed over the same polyhedral partition of the domain. However, if the convex hull of the graph of $h(\boldsymbol{z})$ over $Q_{j,i}$ depends only on function value at the vertices of $Q_{j,i}$, as is the case when $h(\boldsymbol{z})$ is multilinear and the partition is hyper-rectangular, the smallest PPR (SPPR) can be constructed by choosing $\bar{Q}_i = \operatorname{conv}\{(\boldsymbol{z}, h(\boldsymbol{z}))\}_{\boldsymbol{z}\in \operatorname{vert} Q_i}$ for all $i \in [L]$. When $L = 1$, the SPPR of a multilinear function h describes the convex hull of the graph of h over its hyper-rectangular domain; see [30, 3, 18] for convex hull results for multilinear functions. Given a relaxation of a nonlinear function defined over a polyhedral partition, a tighter relaxation can typically be obtained by subdividing the original partition. We refer to a relaxation of an optimization problem Σ obtained by relaxing nonlinear or nonconvex functions using PPRs as a *PPR of* Σ.

In this paper, we consider the PPR of (1.1) obtained by individually modeling the SPPR of each f_j over an HP, since it is commonly used in software implementations. Each function f_j is defined over a subset of the variables $\boldsymbol{z}$, which sometimes will substantially overlap with (or even exactly match) the variables appearing in other functions $f_{j'}$ of the collection. When this happens, we might choose to use a common HP for the domain of all variables appearing in these functions. More precisely, we assume that we are given n_Q HPs. For all $t \in [n_Q]$, the t^{th} HP denoted by $\{Q_{t,i}\}_{i\in[L_t]}$ is an HP of $\mathcal{Z}_{J_t}$ where $J_t \subseteq [n]$. A function $\sigma : [n_w] \mapsto [n_Q]$ is given to indicate which HP is used to define the SPPR of f_j. Clearly, it is required that $I_j \subseteq J_{\sigma(j)}$ for all $j \in [n_w]$. For the purpose of simplicity, we assume without loss of generality (WLOG) that the domain space of f_j and the dimensional space of the $\sigma(j)^{\text{th}}$ HP are same, *i.e.*, $I_j = J_{\sigma(j)}$ for all $j \in [n_w]$ throughout this paper. We also assume that σ is surjective since otherwise, at least one of the HPs is not utilized in the construction of the PPR. It follows that $n_Q \leq n_w$. In this paper, we consider the following PPR of (1.1):

$$\text{(1.2a)} \qquad \max \quad \boldsymbol{c}_{\boldsymbol{z}}^{\intercal}\boldsymbol{z} + \boldsymbol{c}_{\boldsymbol{w}}^{\intercal}\boldsymbol{w}$$

$$\text{(1.2b)} \qquad \text{s.t.} \quad A_{\boldsymbol{z}}\boldsymbol{z} + A_{\boldsymbol{w}}\boldsymbol{w} \leq \boldsymbol{b},$$

$$\text{(1.2c)} \qquad \begin{pmatrix} \boldsymbol{z}_{I_j} \\ w_j \end{pmatrix} \in \bigcup_{i\in[L_{\sigma(j)}]} \operatorname{conv}\left\{ \begin{pmatrix} \bar{\boldsymbol{z}} \\ f_j(\bar{\boldsymbol{z}}) \end{pmatrix} \right\}_{\bar{\boldsymbol{z}}\in Q_{\sigma(j),i}}, \qquad \forall j \in [n_w].$$

We refer to (1.2) as an individual piecewise polyhedral relaxation (IPPR) of (1.1). In (1.2), we relaxed each function f_j independently. Individual relaxation is a common technique to obtain relaxations of nonlinear optimization problems. McCormick's relaxation [19] is a frequently-used such example. In this paper, we seek to derive constraints that improve individual relaxation-based formulations by taking advantage of the connections they share.

We say that a HP $\{Q_i\}_{i\in[L]}$ of a hyper-rectangle $S \subseteq \mathbb{R}^n$ is *regular* (or equivalently that $\{Q_i\}_{i\in[L]}$ is a *regular hyper-rectangular partition (RHP)*) if, for all coordinate axes $v \in [n]$, there exists a collection of intervals $\delta_{v,k} := [d_{v,k}, d_{v,k+1}]$ for $k \in [D_v - 1]$, where $d_{v,1}, \ldots, d_{v,D_v} \in \mathbb{R}$ are sorted in increasing order, such that each element Q_i is of the form $\prod_{v\in[n]} \delta_{v,k_{v,i}}$ for some $k_{v,i} \in [D_v - 1]$ for all $v \in [n]$. Formulations of piecewise approximations in the literature [31, 15, 22, 28] are often based on RHPs, which may further be refined to be simplicial.

A mixed-integer linear programming (MILP) formulation is said to be *ideal* if every extreme point of its LP relaxation complies with the corresponding integrality requirement of the formulation. A formulation for (1.2) is said to be *locally ideal* if it is ideal when there is one HP in the problem, *i.e.*, $n_Q = 1$.

The theoretical contributions of this paper include (i) valid inequalities for (1.1) that tighten MILP formulations for (1.2), and (ii) the first locally ideal MILP formulations for (1.2) over (non-regular) HPs. The tightening exploits the shared variables across HPs by interpreting convex multipliers of vertices of $Q_{t,i}$ as multilinear expressions. This interpretation also makes it possible to use recent developments on relaxations for multilinear optimization problems in the construction of the locally-ideal formulations mentioned above [16]. The paper also makes computational contributions by showing that ALPINE, an open-source global solver for MINLPs, can be enhanced with these formulations so as to substantially improve its computational performance.

2. MILP Formulations Over RHPs.

In this section, we consider (1.2) over RHPs. We assume that RHPs $\{Q_{t,i}\}_{i\in[L_t]}$ for all $t \in [n_Q]$ share common discretization points on their axes. It follows that $\sigma(j) = \sigma(j')$ if $I_j = I_{j'}$ for all $j, j' \in [n_w]$. This assumption is prevalent in the literature; see [31, 28].

We provide a new MILP formulation for (1.2) that we prove is locally ideal and does not require too many variables. Moreover, we develop a mixed-integer multilinear programming formulation (MLP) for (1.1) and, via its linearization, further tighten the proposed MILP formulation. Finally, we perform computational experiments by integrating some of our results into ALPINE, since this code uses PPRs to obtain bounds when solving MINLPs to global optimality.

2.1. A Locally Ideal MILP Formulation Over RHPs.

We first describe an MILP formulation for (1.2), which we refer to as (IPPR1). We use five types of decision variables: $\boldsymbol{z}$, $\boldsymbol{w}$, $\boldsymbol{\lambda}$, $\boldsymbol{\rho}$, and $\boldsymbol{x}$. Variables $\boldsymbol{z}$ and $\boldsymbol{w}$ are the same variables used in (1.1). Binary variable $x_{v,j}$ indicates an interval $\delta_{v,j}$ that contains the value of z_v for all $v \in [n]$ and for all $j \in [D_v - 1]$. Variable $\boldsymbol{\lambda}$ represents the convex combination weights used to express the values of $\boldsymbol{w}$. For $I \subseteq [n]$, we denote by $\bar{\mathcal{Z}}_I := \prod_{v\in I}\{d_{v,k}\}_{k\in[D_v]}$ the set of all vertices that can be used in convex combinations in the space of $\boldsymbol{z}_I$. Variable $\lambda^{\boldsymbol{z}_I}_{\bar{\boldsymbol{z}}}$ indicates the convex combination weight for vertex $\bar{\boldsymbol{z}}$ in the space of $\boldsymbol{z}_I$ for all $I \in \mathcal{I}$ and for all $\bar{\boldsymbol{z}} \in \mathcal{Z}_I$ where $\mathcal{I} := \{I \subseteq [n] \mid \exists j \in [n_w] : I = I_j\}$ is the collection of all nonduplicate sets I_j. Variable $\rho_{v,k,\mathrm{lb}}$ (resp. $\rho_{v,k,\mathrm{ub}}$) represents the accumulated convex combination weight on the lower-bound (resp. upper-bound) of interval $\delta_{v,k}$ on the z_v-axis when $x_{v,k} = 1$ for all $v \in [n]$ and for all $k \in [D_v - 1]$. For $I \subseteq [n]$, $\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_I$, and $v \in I$, we denote by $\bar{\boldsymbol{z}}_{(v)}$ the component corresponding to z_v in $\bar{\boldsymbol{z}}$. For a positive integer K, we use $\Delta^K := \{\boldsymbol{x} \in \mathbb{R}^K_+ \mid \sum_{j\in[K]} x_j = 1\}$ to denote the simplex having as vertices the K principal vectors of $\mathbb{R}^K$ and use $\Delta_{0,1}^K$ to denote its vertices. Finally, we use the convention that $\rho_{v,0,\mathrm{ub}} = \rho_{v,D_v,\mathrm{lb}} = 0$ for all $I \in \mathcal{I}$ and for all $v \in I$. (IPPR1) can then be described as follows:

$$
\begin{aligned}
&\text{(2.1a)} && \max && \boldsymbol{c}_{\boldsymbol{z}}^\intercal \boldsymbol{z} + \boldsymbol{c}_{\boldsymbol{w}}^\intercal \boldsymbol{w} && \\
&\text{(2.1b)} && \text{s.t.} && A_{\boldsymbol{z}} \boldsymbol{z} + A_{\boldsymbol{w}} \boldsymbol{w} \le \boldsymbol{b}, && \\
&\text{(2.1c)} && && z_v = \sum_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_I} \bar{\boldsymbol{z}}_{(v)} \lambda_{\bar{z}}^{\boldsymbol{z}_I}, && \forall I \in \mathcal{I}, \forall v \in I, \\
&\text{(2.1d)} && && w_j = \sum_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_{I_j}} f_j(\bar{\boldsymbol{z}}) \lambda_{\bar{z}}^{\boldsymbol{z}_{I_j}}, && \forall j \in [n_w], \\
&\text{(2.1e)} && && \sum_{\bar{z} \in \bar{\mathcal{Z}}_I : \bar{z}_{(v)} = d_{v,k}} \lambda_{\bar{z}}^{\boldsymbol{z}_I} = \rho_{v,k-1,\mathrm{ub}} + \rho_{v,k,\mathrm{lb}} && \forall I \in \mathcal{I}, \forall v \in I, \forall k \in [D_v], \\
&\text{(2.1f)} && && \rho_{v,k,\mathrm{lb}} + \rho_{v,k,\mathrm{ub}} \le x_{v,k} && \forall v \in [n], \forall k \in [D_v - 1], \\
&\text{(2.1g)} && && \lambda_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_I} \ge 0, && \forall I \in \mathcal{I}, \forall \bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_I, \\
&\text{(2.1h)} && && \boldsymbol{\rho_v} = \{\rho_{v,k,a}\}_{k \in [D_v - 1], a \in \{\mathrm{lb},\mathrm{ub}\}} \in \Delta^{2D_v - 2}, && \forall v \in [n], \\
&\text{(2.1i)} && && \boldsymbol{x_v} \in \Delta_{0,1}^{D_v - 1}, && \forall v \in [n].
\end{aligned}
$$

(IPPR1) models (1.2), *i.e.*, its projection in the space of $(\boldsymbol{z}, \boldsymbol{w})$ is equal to (1.2). We precisely show in Proposition 2.1 that (IPPR1) is a relaxation of (1.1). To this end, we first introduce a mixed-integer multilinear formulation for (1.1), which we denote by (MLP), that uses the same variables as in (2.1) except $\boldsymbol{\lambda}$. Then, we show that (IPPR1) can be obtained by linearizing (MLP). This, in turn, proves that (IPPR1) is a relaxation of (1.1). Model (MLP) is as follows:

$$
\begin{aligned}
&\text{(2.2a)} && \max && \boldsymbol{c}_{\boldsymbol{z}}^\intercal \boldsymbol{z} + \boldsymbol{c}_{\boldsymbol{w}}^\intercal \boldsymbol{w} && \\
&\text{(2.2b)} && \text{s.t.} && A_{\boldsymbol{z}} \boldsymbol{z} + A_{\boldsymbol{w}} \boldsymbol{w} \le \boldsymbol{b}, && \\
&\text{(2.2c)} && && z_v = \sum_{k \in [D_v]} d_{v,k} (\rho_{v,k-1,\mathrm{ub}} + \rho_{v,k,\mathrm{lb}}), && \forall v \in [n], \\
&\text{(2.2d)} && && w_j = f_j(\boldsymbol{z}_{I_j}), && \forall j \in [n_w], \\
&\text{(2.2e)} && && \rho_{v,k,\mathrm{lb}} + \rho_{v,k,\mathrm{ub}} \le x_{v,k} && \forall v \in [n], \forall k \in [D_v - 1], \\
&\text{(2.2f)} && && \boldsymbol{\rho_v} = \{\rho_{v,k,a}\}_{k \in [D_v - 1], a \in \{\mathrm{lb},\mathrm{ub}\}} \in \Delta^{2D_v - 2}, && \forall v \in [n], \\
&\text{(2.2g)} && && \boldsymbol{x_v} \in \Delta_{0,1}^{D_v - 1}, && \forall v \in [n].
\end{aligned}
$$

The projection in the space of $(\boldsymbol{z}, \boldsymbol{w})$ variables of the feasible set of (MLP) is exactly the same as the feasible set of (1.1). This is because projecting the subsystem of (2.2c) and (2.2e)–(2.2g) onto $\boldsymbol{z}$ yields $\mathcal{Z}$. We show in Proposition 2.1 that any feasible solution of (MLP) can be mapped to a feasible solution of (IPPR1) that has the same values for $\boldsymbol{z}$ and $\boldsymbol{w}$, which proves that (IPPR1) is a relaxation of (1.1).

PROPOSITION 2.1. *Given a feasible solution* $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{w}}, \hat{\boldsymbol{\rho}}, \hat{\boldsymbol{x}})$ *of (MLP), define*

$$
\hat{\lambda}_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_I} := \prod_{v \in I} \hat{\lambda}_{\bar{\boldsymbol{z}}_{(v)}}^{z_v}, \qquad \forall I \in \mathcal{I}, \ \forall \bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_I, \tag{2.3}
$$

where $\hat{\lambda}_{d_{v,k}}^{z_v} := \hat{\rho}_{v,k-1,ub} + \hat{\rho}_{v,k,lb}$ *for all* $v \in [n]$ *and for all* $k \in [D_v]$. *Then,* $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{w}}, \{\hat{\lambda}_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_I}\}_{I \in \mathcal{I}, \bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_I}, \hat{\boldsymbol{\rho}}, \hat{\boldsymbol{x}})$ *is feasible to (IPPR1).*

Proof. It is sufficient to verify that the constraints containing $\{\hat{\lambda}^{\boldsymbol{z}_I}_{\bar{\boldsymbol{z}}}\}_{I\in\mathcal{I},\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_I}$ variables, *i.e.*, (2.1c)–(2.1e) and (2.1g), are satisfied. By construction, $\hat{\lambda}^{\boldsymbol{z}_I}_{\bar{\boldsymbol{z}}} \geq 0$ for all $I \in \mathcal{I}$ and for all $\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_I$, *i.e.*, (2.1g) is satisfied.

We next show that the constructed solution for (IPPR1) satisfies (2.1c)–(2.1e). To do so, we provide an equality relation (2.4) which converts an affine expression of $\{\hat{\lambda}^{\boldsymbol{z}_I}_{\bar{\boldsymbol{z}}}\}_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_I}$ to a multlinear expression of $\{\hat{\lambda}^{z_v}_{d_{v,k}}\}_{v\in I,k\in[D_v]}$ for all $I \in \mathcal{I}$ and for all $v \in I$. Consider $I \in \mathcal{I}$ and, WLOG, assume that $I = [|I|]$. Given a collection of functions $\{g_v(z_v)\}_{v\in I}$ where $g_v(z_v) : [d_{v,1}, d_{v,D_v}] \mapsto \mathbb{R}$ is piecewise-linear in z_v with breakpoints at $\{d_{v,k}\}_{k\in[D_v]}$, the following relation holds:

$$
\begin{aligned}
(2.4)\quad \sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_I}\left(\prod_{v\in I} g_v(\bar{\boldsymbol{z}}_{(v)})\right)\hat{\lambda}^{\boldsymbol{z}_I}_{\bar{\boldsymbol{z}}} &= \sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_I}\prod_{v\in I} g_v(\bar{\boldsymbol{z}}_{(v)})\hat{\lambda}^{z_v}_{\bar{\boldsymbol{z}}_{(v)}} \\
&= \sum_{k_1\in[D_1]}\cdots\sum_{k_s\in[D_{|I|}]}\prod_{v\in I} g_v(d_{v,k_v})\hat{\lambda}^{z_v}_{d_{v,k_v}} = \prod_{v\in I}\sum_{k\in[D_v]} g_v(d_{v,k})\hat{\lambda}^{z_v}_{d_{v,k}},
\end{aligned}
$$

where the first equality holds by definition, the second is obtained by expanding $\bar{\mathcal{Z}}_I$ into its elements, and the last is obtained by factoring out the terms not under the control of each sum.

We show that (2.1e) is satisfied for $I \in \mathcal{I}$, $v' \in I$, and $k' \in [D_v]$ using (2.4) by defining, for all $v \in I$ and for all $k \in [D_v]$, $g_v(d_{v,k}) = 1$ if $v \neq v'$ or $k = k'$ and $g_v(d_{v,k}) = 0$ otherwise. We write

$$
\begin{aligned}
(2.5)\quad \sum_{\bar{z}\in\bar{\mathcal{Z}}_I:\bar{z}_{(v')}=d_{v',k'}}\hat{\lambda}^{\boldsymbol{z}_I}_{\bar{z}} &= \sum_{\bar{z}\in\bar{\mathcal{Z}}_I}\left(\prod_{v\in I} g_v(\bar{\boldsymbol{z}}_{(v)})\right)\hat{\lambda}^{\boldsymbol{z}_I}_{\bar{z}} = \prod_{v\in I}\sum_{k\in[D_v]} g_v(d_{v,k})\hat{\lambda}^{z_v}_{d_{v,k}} \\
&= \sum_{k\in[D_{v'}]} g_{v'}(d_{v',k})\hat{\lambda}^{z_{v'}}_{d_{v',k}} = \hat{\lambda}^{z_{v'}}_{d_{v',k'}} = \hat{\rho}_{v',k'-1,\text{ub}} + \hat{\rho}_{v',k',\text{lb}},
\end{aligned}
$$

where the first equality holds by the definition of $\{g_v(z_v)\}_{v\in I}$, the second by (2.4), the third because $\sum_{k\in[D_v]} g_v(d_{v,k})\hat{\lambda}^{z_v}_{d_{v,k}} = \sum_{k\in[D_v]}\hat{\lambda}^{z_v}_{d_{v,k}} = 1$ for $v \neq v'$, the fourth by the definition of $g_{v'}(z_{v'})$, and the last by the definition of $\hat{\lambda}^{z_{v'}}_{d_{v',k'}}$ given in the proposition statement.

We next show that (2.1c) is satisfied. Consider $I \in \mathcal{I}$ and $S \subseteq I$. Define, for all $v \in I$ and for all $k \in [D_v]$, $g_v(d_{v,k}) = d_{v,k}$ if $v \in S$ and $g_v(d_{v,k}) = 1$ otherwise. Then, the following relation holds:

$$
\begin{aligned}
(2.6)\quad \sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_I}\left(\prod_{v\in S}\bar{\boldsymbol{z}}_{(v)}\right)\hat{\lambda}^{\boldsymbol{z}_I}_{\bar{z}} &= \sum_{\bar{z}\in\bar{\mathcal{Z}}_I}\left(\prod_{v\in I} g_v(\bar{\boldsymbol{z}}_{(v)})\right)\hat{\lambda}^{\boldsymbol{z}_I}_{\bar{z}} = \prod_{v\in I}\sum_{k\in[D_v]} g_v(d_{v,k})\hat{\lambda}^{z_v}_{d_{v,k}} \\
&= \prod_{v\in S}\sum_{k\in[D_v]} d_{v,k}\hat{\lambda}^{z_v}_{d_{v,k}} = \prod_{v\in S}\sum_{k\in[D_v]} d_{v,k}(\hat{\rho}_{v,k-1,\text{ub}} + \hat{\rho}_{v,k,\text{lb}}) = \prod_{v\in S}\hat{z}_v,
\end{aligned}
$$

where the first steps follow closely those of (2.5) and the last step holds by (2.2c). Applying (2.6) for all $I \in \mathcal{I}$ and for all $S = \{v\} \subseteq I$, we show that (2.1c) is satisfied for all $I \in \mathcal{I}$ and for all $v \in I$.

Finally, we show that (2.1d) is satisfied for $j \in [n_w]$. Multilinear function $f_j(z_{I_j})$ can be written as $\sum_{S\subseteq I_j}\alpha_S\prod_{v\in S} z_v$ for suitable coefficients $\alpha_S \in \mathbb{R}$ for all $S \subseteq I_j$. Then,

$$\sum_{\bar{z}\in\bar{\mathcal{Z}}_{I_j}} f_j(\bar{z})\hat{\lambda}_{\bar{z}}^{z_{I_j}} = \sum_{\bar{z}\in\bar{\mathcal{Z}}_{I_j}} \left(\sum_{S\subseteq I_j}\alpha_S\prod_{v\in S}\bar{z}_{(v)}\right)\hat{\lambda}_{\bar{z}}^{z_{I_j}}$$
$$= \sum_{S\subseteq I_j}\alpha_S\left(\sum_{\bar{z}\in\bar{\mathcal{Z}}_{I_j}}\left(\prod_{v\in S}\bar{z}_{(v)}\right)\hat{\lambda}_{\bar{z}}^{z_{I_j}}\right) = \sum_{S\subseteq I_j}\alpha_S\prod_{v\in S}\hat{z}_v = f_j(\hat{z}_{I_j}) = \hat{w}_j,$$

where the first and fourth equalities are obtained by using the expression of $f_j(z_{I_j})$, the second is obtained by switching the order of summations, the third is obtained by (2.6), and the last holds by (2.2d). ∎

We next show that (IPPR1) is locally ideal and does not require too many variables.

THEOREM 2.2. *(IPPR1) is a locally ideal formulation for (1.2) and its size (the total number of variables and constraints) is polynomial in the total number of variables and constraints in (1.1) and in the total number of vertices used in convex combination expressions, i.e.,* $\sum_{I\in\mathcal{I}}|\bar{\mathcal{Z}}_I|$.

Proof. The statement about the size of (IPPR1) is clear. We thus only show that (IPPR1) is locally ideal. Consider (IPPR1) with $n_Q = 1$. It follows that $\mathcal{I} = \{I_1\}$. We denote by S the feasible set of (IPPR1). To this end, we construct set S_0 by selecting all the constraints of (IPPR1) only containing variables $(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}^{z_{I_1}}, \boldsymbol{\rho})$ where $\boldsymbol{\lambda}^{z_{I_1}} = \{\lambda_{\bar{z}}^{z_{I_1}}\}_{\bar{z}\in\bar{\mathcal{Z}}_{I_1}}$. For all $v \in [n]$, we construct set S_v by selecting all the constraints of (IPPR1) only containing variables $(\boldsymbol{\rho_v}, \boldsymbol{x_v})$. Then, we can reformulate S as

$$S = \left\{(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}^{z_{I_1}}, \boldsymbol{\rho}, \boldsymbol{x}) \;\middle|\; \begin{array}{ll} (\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}^{z_{I_1}}, \boldsymbol{\rho}) \in S_0, & \\ (\boldsymbol{\rho_v}, \boldsymbol{x}_v) \in S_v, & \forall v \in [n] \end{array}\right\}.$$

Observe that S_0 and S_v share $\boldsymbol{\rho_v}$ for all $v \in [n]$ and $S_{v'}$ and $S_{v''}$ do not share any variable for $v' \neq v'' \in [n]$. Moreover, $\boldsymbol{\rho_v}$ forms a simplex for all $v \in [n]$. It follows that $\text{conv}(S)$ can be obtained by separately convexifying S_i for $i \in \{0, \ldots, n\}$; see [27] or [16]. We obtain that

$$\text{(2.7)}\qquad \text{conv}(S) = \left\{(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}^{z_{I_1}}, \boldsymbol{\rho}, \boldsymbol{x}) \;\middle|\; \begin{array}{ll} (\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}^{z_{I_1}}, \boldsymbol{\rho}) \in \text{conv}(S_0), & \\ (\boldsymbol{\rho_v}, \boldsymbol{x}_v) \in \text{conv}(S_v), & \forall v \in [n] \end{array}\right\}.$$

The constraints that belong to S_0 describe $\text{conv}(S_0)$ because there is no integral requirement in S_0, *i.e.*, $S_0 = \text{conv}(S_0)$. Moreover, for all $v \in [n]$, S_v is described by the system of (2.1f), (2.1h), and (2.1i), which is referred to as the disaggregated convex combination formulation in [31] and this formulation is known to be ideal. In conclusion, for each $i \in \{0, \ldots, n\}$, (IPPR1) contains all the constraints that describe $\text{conv}(S_i)$. It follows that (IPPR1) is locally ideal by (2.7). ∎

We remark that (IPPR1) is a new locally ideal formulation for (1.2). Various other MILP formulations can be obtained by viewing the set of expressions, $E_{I,v} = \{\sum_{\bar{z}\in\bar{\mathcal{Z}}_I:\bar{z}_{(v)}=d_{v,k}}\lambda_{\bar{z}}^{z_I}\}_{k\in[D_v]}$, as special ordered sets of type 2 (SOS2) for all $I \in \mathcal{I}$ and for all $v \in I$. For example, another locally ideal MILP formulation is obtained by removing $\boldsymbol{\rho}$ together with all the constraints containing $\boldsymbol{\rho}$, *i.e.*, (2.1e), (2.1f), and (2.1h), from (IPPR1) and by adding (2.8):

$$\text{(2.8a)}\qquad \sum_{\bar{z}\in\bar{\mathcal{Z}}_I:\bar{z}_{(v)}\leq d_{v,\ell}}\lambda_{\bar{z}}^{z_I} \leq \sum_{k\in[\ell]} x_{v,k}, \qquad \forall I\in\mathcal{I}, \forall v\in I, \forall \ell\in[D_v-2],$$

$$\sum_{k\in[\ell]} x_{v,k} \le \sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_I:\bar{\boldsymbol{z}}_{(v)}\le d_{v,\ell+1}} \lambda^{\boldsymbol{z}_I}_{\bar{\boldsymbol{z}}}, \qquad \forall I\in\mathcal{I}, \forall v\in I, \forall \ell\in[D_v-2], \tag{2.8b}$$

where (2.8) is inspired from a locally ideal formulation for SOS2 [17]. We can prove that the obtained formulation is locally ideal using the same decomposition technique as in the proof of Theorem 2.2. However, introducing $\boldsymbol{\rho}$ in (IPPR1) has the advantage of creating a formulation that is sparser than the formulation with (2.8). Specifically, a single $\lambda^{\boldsymbol{z}_I}_{\bar{\boldsymbol{z}}}$ variable appears n times in (2.1e), whereas it may appear at most $n(K-1)$ times in (2.8) where $K := \max_{v\in[n]} D_v$. Given that the number of variables $\boldsymbol{\lambda}$ is large (at most $\sum_{I\in\mathcal{I}} K^{|I|}$), the sparsity of (IPPR1) could prove to be useful for numerical computations. We leave this avenue of research for future work and instead focus our computational work on exploring the improved tightness obtained from the linking constraints discussed in section 2.2. To clearly evaluate the effect of the linking constraints, we will use the sharp formulation for (1.2) implemented in ALPINE [29] and will not introduce the proposed $\boldsymbol{\rho}$ variables.

2.2. Linking Constraints. Inspired from (MLP), we next introduce linear equalities in Proposition 2.4, which we call *linking constraints*, that can be added to (IPPR1) to make it a tighter relaxation of (1.1). To streamline the presentation, we define $\lambda^{z_v}_{d_{v,k}} := \rho_{v,k-1,\text{ub}} + \rho_{v,k,\text{lb}}$ for all $v\in[n]$ and for all $k\in[D_v]$, use $\boldsymbol{\lambda}^{z_v} = \{\lambda^{z_v}_{d_{v,k}}\}_{k\in[D_v]}$ for all $v\in[n]$, and use $\boldsymbol{\lambda}^{\boldsymbol{z}_I} = \{\lambda^{\boldsymbol{z}_I}_{\bar{\boldsymbol{z}}}\}_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_I}$ for all $I\in\mathcal{I}$.

We next motivate these equalities as providing relationships between degree-$|S|$ multilinear terms of $\{\boldsymbol{\lambda}^{z_v}\}_{v\in S}$ and degree-$|T|$ multilinear terms of $\{\boldsymbol{\lambda}^{z_v}\}_{v\in T}$, for given $S\subsetneq T\subseteq[n]$. Let $S=\{v_1,\dots,v_s\}$ and $T=\{v_1,\dots,v_t\}$ with $t>s$. For $k_1\in[D_{v_1}]$, ..., $k_s\in[D_{v_s}]$, it holds that

$$\prod_{i=1}^{s}\lambda^{z_{v_i}}_{d_{v_i,k_i}} = \left(\prod_{i=1}^{s}\lambda^{z_{v_i}}_{d_{v_i,k_i}}\right)\prod_{i=s+1}^{t}\left(\sum_{k\in[D_{v_i}]}\lambda^{z_{v_i}}_{d_{v_i,k}}\right) = \sum_{k_{s+1}=1}^{D_{s+1}}\cdots\sum_{k_t=1}^{D_t}\prod_{i=1}^{t}\lambda^{z_{v_i}}_{d_{v_i,k_i}}, \tag{2.9}$$

where the first equality holds by definition and (2.2f), and the second is obtained by expanding the expression. Using (2.3) and (2.9), we obtain the following constraints that link $\boldsymbol{\lambda}^{\boldsymbol{z}_{T_1}}$ and $\boldsymbol{\lambda}^{\boldsymbol{z}_{T_2}}$:

$$\sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_{T_1}:\hat{\boldsymbol{z}}=\text{proj}_{\boldsymbol{z}_S}\bar{\boldsymbol{z}}}\lambda^{\boldsymbol{z}_{T_1}}_{\bar{\boldsymbol{z}}} = \sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_{T_2}:\hat{\boldsymbol{z}}=\text{proj}_{\boldsymbol{z}_S}\bar{\boldsymbol{z}}}\lambda^{\boldsymbol{z}_{T_2}}_{\bar{\boldsymbol{z}}}, \quad \forall S\subseteq(T_1\cap T_2):|S|>0,\ \forall\hat{\boldsymbol{z}}\in\bar{\mathcal{Z}}_S \tag{2.10}$$

for $T_1,T_2\in\mathcal{I}$. The condition $|S|>0$ implies $T_1\cap T_2\neq\emptyset$ in these equalities. Both sides of (2.10) correspond to degree-$|S|$ multilinear term $\prod_{v\in S}\lambda_{v,k_v}$ where $k_v\in[D_v]$ is such that $d_{v,k_v}=\hat{\boldsymbol{z}}_{(v)}$ for $v\in S$.

Linking relation (2.10) also can be naturally observed through a reformulation-linearization technique presented in [13]. The idea is to remove $\rho_{v,D_v,\text{lb}}$ from (MLP) by plugging $\rho_{v,D_v,\text{lb}} = 1-(\mathbf{1}^\intercal\boldsymbol{\rho_v}-\rho_{v,D_v,\text{lb}})$ for all $v\in[n]$ so that the remaining $\boldsymbol{\rho}$ variables are all independent. Then, for some $I\subseteq[n]$, $\prod_{v\in I}x_v$ is written as

$$\prod_{v\in I}x_v = \sum_{\{v_1,\dots,v_p\}\subseteq I}\ \sum_{k_1\in[D_{v_1}-1]}\cdots\sum_{k_p\in[D_{v_p}-1]}\ \prod_{i\in[p]}(d_{v_i,k_i}-d_{v_i,D_{v_i}})\lambda^{z_{v_i}}_{d_{v_i,k_i}}, \tag{2.11}$$

when plugging (2.2c). Expression (2.11) contains multilinear terms consisting of $\{\lambda^{z_v}_{d_{v,k}}\}_{v\in I,k\in[D_v-1]}$ of degree from 1 to $|I|$. Considering $j_1,j_2\in[n_w]$ such that $I_{j_1}\neq$

I_{j_2} and $I_{j_1} \cap I_{j_2} \neq \emptyset$, (2.2d) for $j \in \{j_1, j_2\}$ have common multilinear terms with degree up to $|I_{j_1} \cap I_{j_2}|$, which are $\prod_{i\in[p]} \lambda^{z_{v_i}}_{d_{v_i},k_i}$ for all non-empty set $\{v_1, \ldots, v_p\} \subseteq I_{j_1} \cap I_{j_2}$ and for all $(k_1, \ldots, k_p) \in \prod_{i\in[p]}[D_{v_i} - 1]$. It follows that the expressions of w_{j_1} and w_{j_2} after linearization of the formulation with independent $\boldsymbol{\rho}$ have common λ variables, *i.e.*, it implies (2.10).

THEOREM 2.3. *Consider formulation (IPPR1), i.e., all the variables and constraints in* (2.1)*, together with additional variable* $\{\lambda^{\boldsymbol{z}_S}_{\hat{\boldsymbol{z}}}\}_{S\in\mathcal{S},\hat{\boldsymbol{z}}\in\bar{\mathcal{Z}}_S}$*, where* $\mathcal{S} = \{S \subseteq [n] \setminus \{\emptyset\} \mid \exists T_1 \neq T_2 \in \mathcal{I} : S \subsetneq T_1 \cap T_2\}$ *and the following constraints*

$$\lambda^{\boldsymbol{z}_S}_{\hat{\boldsymbol{z}}} = \sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_T:\hat{\boldsymbol{z}}=\mathrm{proj}_{\boldsymbol{z}_S}\bar{\boldsymbol{z}}} \lambda^{\boldsymbol{z}_T}_{\bar{\boldsymbol{z}}}, \qquad \forall (S,T) \in \mathcal{S}\times\mathcal{I} : S \subsetneq T,\ \forall \hat{\boldsymbol{z}} \in \bar{\mathcal{Z}}_S. \tag{2.12}$$

We refer this formulation as (LINK1). Denote by ω_1*,* ω_2*, and* ω_3 *the optimal objective values of (MLP), (LINK1), and (IPPR1), respectively. Then,* $\omega_1 \leq \omega_2 \leq \omega_3$*.*

Proof. Clearly, $\omega_2 \leq \omega_3$. Also, it holds that $\omega_1 \leq \omega_2$ because (2.12) is equivalent to (2.10), which is valid to (MLP) by (2.3) in Proposition 2.1 and (2.9). ∎

The feasible set of (LINK1) can be strictly contained in that of (IPPR1) as we will illustrate in Example 1. The numbers of additional variables and constraints, however, can become large when there are many discretization points on each axis, as a different constraint is imposed for every element of $\bar{\mathcal{Z}}_S$ and for each $(S,T) \in \mathcal{S}\times\mathcal{I}$ such that $S \subsetneq T$. In Proposition 2.4, we show that, after adding suitable variables, we can reduce the number of constraints necessary to one for each such (S,T). The constraint in question is obtained by aggregating $|\bar{\mathcal{Z}}_S|$ constraints with multiplier $\prod_{v\in S} \hat{\boldsymbol{z}}_{(v)}$ for all $\hat{\boldsymbol{z}} \in \bar{\mathcal{Z}}_S$.

PROPOSITION 2.4. *Consider formulation (IPPR1), i.e., all the variables and constraints in* (2.1)*, together with additional variable* μ_S *for all* $S \in \mathcal{S}$ *such that* $|S| \geq 2$*, and the following constraints*

$$\mu_S = \sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_T} \left(\prod_{v\in S} \bar{\boldsymbol{z}}_{(v)}\right) \lambda^{\boldsymbol{z}_T}_{\bar{\boldsymbol{z}}}, \qquad \forall (S,T) \in \mathcal{S}\times\mathcal{I} : S \subsetneq T, |S| \geq 2. \tag{2.13}$$

Every feasible solution of this formulation, which we refer to as (LINK2), satisfies (2.10) *for all* $T_1, T_2 \in \mathcal{I}$ *such that* $T_1 \cap T_2 \neq \emptyset$*.*

Proof. We refer to the cardinality of the set S used in the description of (2.10) or (2.13) as the degree of this inequality. We prove the result by induction. We show that every feasible solution of (LINK2) satisfies all constraints (2.10) of degree 1. Then we show that, if all constraints (2.10) of degree up to d are satisfied, then all of the constraints (2.10) of degree up to $d+1$ are satisfied as well.

We first argue that all constraints (2.10) of degree 1 are satisfied by (IPPR1), which is a relaxation of (LINK2). Pick any index $(S, T_1, T_2, \hat{\boldsymbol{z}})$ of (2.10) of degree 1, *i.e.*, $T_1, T_2 \in \mathcal{I}$, $S \subseteq T_1 \cap T_2$, $\hat{\boldsymbol{z}} \in \bar{\mathcal{Z}}_S$, and $|S| = 1$. We denote the unique element of S by v. The values of the left-hand-side (lhs) and right-hand-side (rhs) of (2.10) can be interpreted as the convex combination weight at $z_v = \hat{\boldsymbol{z}}_{(v)}$ by (2.1c):

$$z_v = \sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_{T_i}} \bar{\boldsymbol{z}}_{(v)} \lambda^{\boldsymbol{z}_{T_i}}_{\bar{\boldsymbol{z}}} = \sum_{j\in[D_v]} d_{v,j} \left(\sum_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_{T_i}:\bar{\boldsymbol{z}}_{(v)}=d_{v,j}} \lambda^{\boldsymbol{z}_{T_i}}_{\bar{\boldsymbol{z}}} \right),$$

for $i \in \{1, 2\}$. Constraints (2.1e)–(2.1i) impose that at most two consecutive weights on axis z_v can be positive. It follows that, given the value of z_v, all weights on axis z_v are uniquely determined since each point on a line segment is a unique convex combination of its ending points. Therefore, for all $j \in [D_v]$, it holds that $\sum_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_{T_1}: \bar{\boldsymbol{z}}_{(v)} = d_{v,j}} \lambda_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_{T_1}} = \sum_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_{T_2}: \bar{\boldsymbol{z}}_{(v)} = d_{v,j}} \lambda_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_{T_2}}$. We conclude that all the constraints (2.10) of degree 1 are implied by (IPPR1).

For any positive integer $d \le \max_{S \in \mathcal{S}} |S| - 1$, we next argue that if (LINK2) implies all constraints (2.10) of degree up to d, then all constraints (2.10) of degree up to $d+1$ are also implied by (LINK2). Consider any such integer d. Assume that all constraints (2.10) of degree up to d are implied by (LINK2). Pick $(S, T_1, T_2) \subseteq \mathcal{S} \times \mathcal{I} \times \mathcal{I}$ such that $S \subseteq T_1 \cap T_2$ and $|S| = d+1$. We show that (2.10) with indices $(S, T_1, T_2, \hat{\boldsymbol{z}})$ are implied by (LINK2) for all $\hat{\boldsymbol{z}} \in \bar{\mathcal{Z}}_S$. Consider a feasible solution $(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ of (LINK2). For all $v \in [n]$, we denote by k_v the index such that $x_{v,k_v} = 1$. We define $\hat{Q} = \prod_{v \in S} \delta_{v,k_v}$. Clearly, $\hat{Q}$ is a hyper-rectangle in the space of $\boldsymbol{z}_S$. By (2.1e)–(2.1i), for $I \in \{T_1, T_2\}$ and for $\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_I$, $\lambda_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_I}$ is zero if $\operatorname{proj}_{\boldsymbol{z}_S} \bar{\boldsymbol{z}} \notin \operatorname{vert} \hat{Q}$. It follows that many constraints (2.10) are satisfied because they reduce to $0 = 0$. The 2^{d+1} constraints (2.10) that do not simplify in $0 = 0$ are those with indices $(S, T_1, T_2, \hat{\boldsymbol{z}})$ such that $\hat{\boldsymbol{z}} \in \operatorname{vert} \hat{Q}$. Consider an edge of $\hat{Q}$ with endpoints $\hat{\boldsymbol{z}}'$ and $\hat{\boldsymbol{z}}''$. Since $\hat{Q}$ is a hyper-rectangle in $\boldsymbol{z}_S$, there exists a unique $v \in S$ such that $\operatorname{proj}_{\boldsymbol{z}_{S \setminus \{v\}}} \hat{\boldsymbol{z}}' = \operatorname{proj}_{\boldsymbol{z}_{S \setminus \{v\}}} \hat{\boldsymbol{z}}''$. It follows that the sum of the lhs (resp. rhs) of (2.10) with indices $(S, T_1, T_2, \hat{\boldsymbol{z}}')$ and $(S, T_1, T_2, \hat{\boldsymbol{z}}'')$ is equal to the lhs (resp. rhs) of (2.10) with $(S \setminus \{v\}, T_1, T_2, \operatorname{proj}_{\boldsymbol{z}_{S \setminus \{v\}}} \hat{\boldsymbol{z}}')$, respectively. Therefore, for every edge of $\hat{Q}$, the aggregation of the two constraints associated with the endpoints of the edge corresponds to (2.10) of degree d, which is satisfied by the inductive hypothesis. We next show in Lemma 2.5 that if the 2^d constraints of degree d associated with edges are all satisfied and one of the 2^{d+1} constraints of degree $d+1$ associated with vertices is satisfied, then other $2^{d+1} - 1$ constraints of degree $d+1$ are all satisfied.

LEMMA 2.5. *Consider a hyper-rectangle Q in $\mathbb{R}^d$ for some positive integer d. Consider two vectors $\boldsymbol{x} = \{x_{\boldsymbol{z}}\}_{\boldsymbol{z} \in \operatorname{vert} Q} \in \mathbb{R}^{2^d}$ and $\boldsymbol{y} = \{y_{\boldsymbol{z}}\}_{\boldsymbol{z} \in \operatorname{vert} Q} \in \mathbb{R}^{2^d}$. Suppose $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy*

$$x_{\boldsymbol{z}_1} + x_{\boldsymbol{z}_2} = y_{\boldsymbol{z}_1} + y_{\boldsymbol{z}_2}, \qquad \forall \boldsymbol{z}_1, \boldsymbol{z}_2 \in \operatorname{vert} Q : [\boldsymbol{z}_1, \boldsymbol{z}_2] \text{ is an edge of } Q. \tag{2.14}$$

Assume finally that $x_{\boldsymbol{z}'} = y_{\boldsymbol{z}'}$ for some $\boldsymbol{z}' \in \operatorname{vert} Q$. Then $x_{\boldsymbol{z}} = y_{\boldsymbol{z}}$ for all $\boldsymbol{z} \in \operatorname{vert} Q$.

Proof. Consider $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{2^d}$ that satisfies (2.14) and for which $x_{\boldsymbol{z}'} = y_{\boldsymbol{z}'}$ for some $\boldsymbol{z}' \in \operatorname{vert} Q$. Define a simple graph $G = (V, E)$ where $V = \operatorname{vert} Q$ and $E = \{(\boldsymbol{z}_1, \boldsymbol{z}_2) \mid \boldsymbol{z}_1, \boldsymbol{z}_2 \in \operatorname{vert} Q, [\boldsymbol{z}_1, \boldsymbol{z}_2] \text{ is an edge of } Q\}$. We denote by $\operatorname{dist}(\boldsymbol{z}_1, \boldsymbol{z}_2)$ the number of edges of a path with fewest edges in G between $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. The maximum distance between two vertices is bounded above by d. We prove that $x_{\boldsymbol{z}} = y_{\boldsymbol{z}}$ for all $\boldsymbol{z} \in \operatorname{vert} Q$ by induction on $\operatorname{dist}(\boldsymbol{z}, \boldsymbol{z}')$.

First, it holds that $x_{\boldsymbol{z}} = y_{\boldsymbol{z}}$ for all $\boldsymbol{z} \in \operatorname{vert} Q$ such that $\operatorname{dist}(\boldsymbol{z}, \boldsymbol{z}') = 0$ by the lemma's last assumption. Next, we assume that $x_{\boldsymbol{z}} = y_{\boldsymbol{z}}$ for all $\boldsymbol{z} \in \operatorname{vert} Q$ such that $\operatorname{dist}(\boldsymbol{z}, \boldsymbol{z}') = k$ for some integer $k \in \{0, 1, \ldots, d-1\}$. We argue that $\boldsymbol{x}_{\boldsymbol{z}} = \boldsymbol{y}_{\boldsymbol{z}}$ for all $\boldsymbol{z} \in \operatorname{vert} Q$ such that $\operatorname{dist}(\boldsymbol{z}, \boldsymbol{z}') = k+1$. Pick any $\boldsymbol{z}_1 \in \operatorname{vert} Q$ with $\operatorname{dist}(\boldsymbol{z}_1, \boldsymbol{z}') = k+1$. Let P be a path with fewest edges in G from $\boldsymbol{z}_1$ to $\boldsymbol{z}'$. denote by $\boldsymbol{z}_2$ the vertex that directly succeeds $\boldsymbol{z}_1$ on P. The inductive hypothesis implies that

$$\boldsymbol{x}_{\boldsymbol{z}_2} = \boldsymbol{y}_{\boldsymbol{z}_2} \tag{2.15}$$

because $\operatorname{dist}(\boldsymbol{z}_2, \boldsymbol{z}') = k$. Further, it holds that

$$\boldsymbol{x}_{\boldsymbol{z}_1} + \boldsymbol{x}_{\boldsymbol{z}_2} = \boldsymbol{y}_{\boldsymbol{z}_1} + \boldsymbol{y}_{\boldsymbol{z}_2} \tag{2.16}$$

because $[\boldsymbol{z}_1, \boldsymbol{z}_2]$ is an edge of Q by the definition of G. Therefore, it follows from (2.15) and (2.16) that $\boldsymbol{x}_{\boldsymbol{z}_1} = \boldsymbol{y}_{\boldsymbol{z}_1}$, which proves the inductive step. ◻

By Lemma 2.5, it is sufficient to show that the solution satisfies one of 2^{d+1} constraints (2.10) of degree $d+1$. By applying an affine transformation if necessary, we may assume that $\hat{Q} = [0,1]^{d+1}$. Then, after removing zero-valued λ variables and zero-coefficient terms, (2.13) with indices (S, T_1) and (S, T_2) reduce to $\mu_S = \lambda_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_{T_1}}$ and $\mu_S = \lambda_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_{T_2}}$, respectively, where $\bar{\boldsymbol{z}} = (1, 1, \dots, 1) \in \mathbb{R}^{d+1}$. It follows that

$$\lambda_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_{T_1}} = \lambda_{\bar{\boldsymbol{z}}}^{\boldsymbol{z}_{T_2}},$$

which is one of the 2^{d+1} constraints (2.10) of degree $d+1$. It follows, by Lemma 2.5, that the solution satisfies all other 2^{d+1} constraints (2.10) of degree $d+1$. This completes the proof of the inductive step. ◻

Example 1 shows an instance of (MLP) for which (IPPR1) has feasible solutions that do not satisfy constraint (2.13).

EXAMPLE 1. *Consider (1.1) with $n = 4$, $n_w = 2$, $n_A = 0$, $\boldsymbol{\ell} = \boldsymbol{0}$, $\boldsymbol{u} = \boldsymbol{1}$, $f_1(z_1, z_2, z_3) = z_1 z_2 z_3$, $f_2(z_2, z_3, z_4) = z_2 z_3 z_4$, $I_1 = \{1, 2, 3\}$, $I_2 = \{2, 3, 4\}$, and arbitrary cost vectors $\boldsymbol{c_z} \in \mathbb{R}^4$ and $\boldsymbol{c_w} \in \mathbb{R}^2$. We assume that there is no partition, i.e., we model an individual polyhedral relaxation for each f_1 and f_2 over their domains $[0,1]^3$. Clearly, $\boldsymbol{\rho}$ and $\boldsymbol{x}$ are not needed in this case. Instead we impose that $\boldsymbol{\lambda}^{(z_1,z_2,z_3)}$ and $\boldsymbol{\lambda}^{(z_2,z_3,z_4)}$ correspond to the vertices of $[0,1]^3$. (IPPR1) for this problem takes the form:*

(2.17a)
$$\max \quad \boldsymbol{c}_{\boldsymbol{z}}^{\intercal} \boldsymbol{z} + \boldsymbol{c}_{\boldsymbol{w}}^{\intercal} \boldsymbol{w}$$

(2.17b)
$$\text{s.t.} \quad z_k = \sum_{(j_1,j_2,j_3) \in \{0,1\}^3} j_k \lambda_{(j_1,j_2,j_3)}^{(z_1,z_2,z_3)}, \qquad \forall k \in \{1,2,3\},$$

(2.17c) $$z_k = \sum_{(j_2,j_3,j_4) \in \{0,1\}^3} j_k \lambda_{(j_2,j_3,j_4)}^{(z_2,z_3,z_4)}, \qquad \forall k \in \{2,3,4\},$$

(2.17d) $$w_1 = \sum_{(j_1,j_2,j_3) \in \{0,1\}^3} (j_1 j_2 j_3) \lambda_{(j_1,j_2,j_3)}^{(z_1,z_2,z_3)} = \lambda_{(1,1,1)}^{(z_1,z_2,z_3)},$$

(2.17e) $$w_2 = \sum_{(j_2,j_3,j_4) \in \{0,1\}^3} (j_2 j_3 j_4) \lambda_{(j_2,j_3,j_4)}^{(z_2,z_3,z_4)} = \lambda_{(1,1,1)}^{(z_2,z_3,z_4)},$$

(2.17f) $$\sum_{(j_1,j_2,j_3) \in \{0,1\}^3} \lambda_{(j_1,j_2,j_3)}^{(z_1,z_2,z_3)} = 1,$$

(2.17g) $$\sum_{(j_2,j_3,j_4) \in \{0,1\}^3} \lambda_{(j_2,j_3,j_4)}^{(z_2,z_3,z_4)} = 1,$$

(2.17h) $$\lambda_{(j_1,j_2,j_3)}^{(z_1,z_2,z_3)} \geq 0, \qquad \forall (j_1,j_2,j_3) \in \{0,1\}^3,$$

(2.17i) $$\lambda_{(j_2,j_3,j_4)}^{(z_2,z_3,z_4)} \geq 0, \qquad \forall (j_2,j_3,j_4) \in \{0,1\}^3.$$

Consider the feasible solution $(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda})$ of (IPPR1) such that $\boldsymbol{z} = (0.5, 0.5, 0.5, 1)$, $\boldsymbol{w} = (0, 0.5)$, $\lambda_{(0,0,1)}^{(z_1,z_2,z_3)} = \lambda_{(1,1,0)}^{(z_1,z_2,z_3)} = \lambda_{(0,0,1)}^{(z_2,z_3,z_4)} = \lambda_{(1,1,1)}^{(z_2,z_3,z_4)} = 0.5$, and all other unspecified λ variables are all zero. Constraint (2.13) for $S = \{2,3\}$ becomes $\mu_{\{2,3\}} =$

$\lambda^{(z_1,z_2,z_3)}_{(0,1,1)} + \lambda^{(z_1,z_2,z_3)}_{(1,1,1)}$ *and becomes* $\mu_{\{2,3\}} = \lambda^{(z_2,z_3,z_4)}_{(1,1,0)} + \lambda^{(z_2,z_3,z_4)}_{(1,1,1)}$, *for* $T = \{1,2,3\}$ *and* $T = \{2,3,4\}$, *respectively. These equalities cannot be satisfied simultaneously by the aforementioned solution, and hence cut it off.*

Assume n_w is polynomially sized in n and the feasible set of (1.2) does not have any linear constraints (1.2b). Further, assume that, for each $j \in [n_w]$, $|I_j|$ is bounded by a constant C. Under these assumptions, it is straightforward to see that each discretized multilinear function can be convexified using $\prod_{v\in I_j} D_v$ convex multipliers, one for each extreme point. However, the simultaneous convex hull of this feasible set could have an exponential extension complexity. To see this, it suffices to consider the special case where $(\ell_v, u_v) = (0,1)$ for all $v \in [n]$, $n_w = \binom{n}{2}$, and each element of n_w multilinear functions is associated with a pair of variables $(z_v, z_{v'})$ so that w_j is defined as $z_v z_{v'}$. Then, the convex hull of the feasible set of (1.2) is the boolean quadric polytope [25] which is known to have an exponential extension complexity [9]. We remark that the linking constraints (2.10) are polynomially many; for each $j \in [n_w]$, they consider multipliers associated with discretization points in a subspace $\boldsymbol{z}_T$ of variables $\boldsymbol{z}_{I_j}$ and there are at most $n_w 2^C$ such subspaces. Therefore, one cannot hope that these linking constraints will, in general, produce the convex hull of the feasible set. Our computations show, however, that the linking constraints help substantially tighten the relaxations.

2.3. Computational Experiments Using Linking Constraints. ALPINE is an open-source MINLP solver that uses PPRs over RHPs. It iteratively solves (1.2) by refining the RHPs with more discretization points. We implement (2.13) inside of ALPINE. Since the formulation used in ALPINE, see [28], uses the same $\boldsymbol{\lambda}$ variables, we only add linking variables and constraints in (LINK2). We refer to this algorithm as `Link`. We refer to the MINLP solvers ALPINE and SCIP [10] with their default settings as `ALPINE` and `SCIP`, respectively.

We consider instances, `mult3` and `mult4` from Los Alamos MINLPLib https://github.com/lanl-ansi/MINLPLib.jl [1] which are collections of multilinear and polynomial optimization problem instances whose nonlinear terms have degrees up to 3 and 4, respectively. We focus on these instances because the proof of Proposition 2.4 establishes that linking constraints are not implied in ALPINE's formulation when there is a multilinear function with degree at least 3 in the optimization problem,

We perform the computational experiments on a computer running Linux Mint 19.3 with Intel i7-6700K CPU cores running at 4.00GHz and 48GB of memory. The code is written in Julia v1.6.3 with JuMP package v0.21.10 [7] and ALPINE package v0.2.7. We use IPOPT v3.13.4 [32] and Gurobi v9.0.3 [11] as the ALPINE's nonlinear and MILP solvers, respectively.

Table 1 displays the number of instances for which the first polyhedral relaxation model of ALPINE-based algorithms, `Link` and `ALPINE`, obtains the optimal objective value as bound. It also presents the average gap of the first relaxation. Table 2 displays the number of instances not solved in an hour and, for those instances solved within an hour, the average solution times using three global algorithms, `Link`, `ALPINE`, and `SCIP`. The computation shows that substantial benefits can be achieved for all types of instances from using linking constraints. In particular, `Link` reduces by a factor 3 the number of instances that cannot be solved within one hour compared to `ALPINE`. Figure 1 displays the performance profile [5] of solution times. Within three minutes, `Link` solves 29% of problem instances, which matches (resp. is three times) the number of instances solved within an hour by `ALPINE` (resp. `SCIP`).

3. MILP Formulations Over Non-Regular HPs. In this section, we introduce MILP formulations for (1.2) for the general case where HPs can be non-regular. We first present in section 3.1 a motivating example that shows that non-regular HPs can be substantially more economical than RHPs in terms of the number of hyper-

Table 1: Tightness of ALPINE's first relaxation models for N instances with same maximum degree (d). N_{Solved} is the number of instances for which Link and ALPINE solve the relaxation within the time limit. $N_{\texttt{Link}}$ (resp. $N_{\texttt{ALPINE}}$) is the number of instances for which the first relaxation model of Link (resp. ALPINE) closes the gap (between the LP relaxation and the optimal objective value.) GapClosed is the average of percentages of gaps closed by Link where ALPINE is used as the basis of comparison among instances that are solved and for which ALPINE does not close the gap.

Type	d	N	N_{Solved}	$N_{\texttt{Link}}$	$N_{\texttt{ALPINE}}$	GapClosed (%)
mult	3	60	60	51	5	99.8
mult	4	60	51	44	3	98.0
poly	3	60	51	19	3	94.3
poly	4	40	20	3	0	89.5
Total		220	182	117	11	96.6

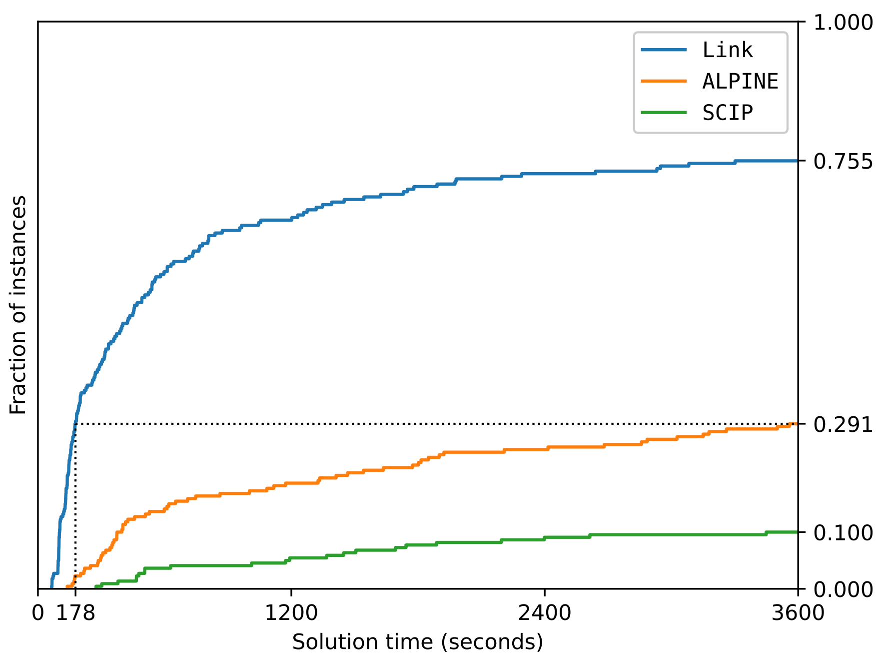


Fig. 1: Performance profile of solution times.

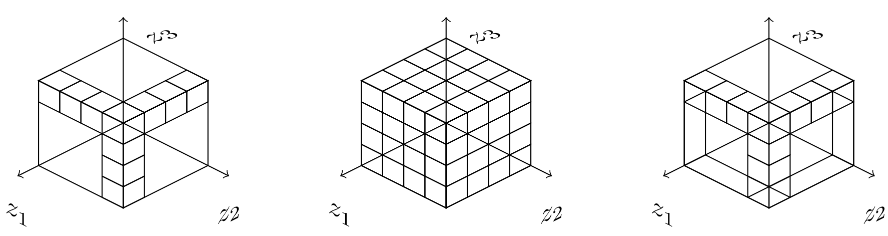
Table 2: Average solution times of N instances with same maximum degree (d). Subscripts represent the number of instances for which the time limit was reached.

Type	d	N	`Link`	`ALPINE`	`SCIP`
mult	3	60	279.2^{1}	2336.8^{33}	3115.7^{46}
mult	4	60	1205.5^{12}	3370.1^{53}	3564.1^{59}
poly	3	60	1226.0^{12}	2473.1^{32}	3261.6^{53}
poly	4	40	3009.0^{29}	3550.9^{38}	3600.0^{40}
Total		220	1286.4^{54}	2876.5^{156}	3365.8^{198}

rectangles they use. In section 3.2, we provide a locally ideal MILP formulation for any PPR of a single (nonlinear) function over a polyhedral partition of the domain. In section 3.3, we introduce new MILP formulations for (1.2) over general HPs that combine the advantages of the formulation described in section 2.1 and the formulation described in section 3.2. Finally, in section 3.5, we perform computational experiments that show that non-regular HPs have discernible computational advantages compared to RHPs.

3.1. Example of an Economical Non-Regular HP. We say that a set of polytopes, $\{Q'_\ell\}_{\ell\in[L']}$, *refines* another set of polytopes, $\{Q_\ell\}_{\ell\in[L]}$, (or equivalently that $\{Q'_\ell\}_{\ell\in[L']}$ is a *refinement* of $\{Q_\ell\}_{\ell\in[L]}$) if, for all $\ell \in [L]$, there exists $S_\ell \subset [L']$ such that $\bigcup_{k\in S_\ell} Q'_k = Q_\ell$. We introduce next in Example 2 a partition that we use in this section to demonstrate that non-regular refinements are more economical than regular refinements.

Example 2. *Let n and K be positive integer parameters. For $v \in [n]$ and for $j \in [K]$, we use $Q_{v,j} = \{\boldsymbol{z} \in \mathbb{R}^n \mid z_v \in [j-1, j], z_{v'} \in [K-1, K], \forall v' \in [n] \setminus \{v\}\}$, which is a unit hyper-cube on an edge of $[0, K]^n$. Consider the partition of $[0, K]^n$, $\mathcal{Q}' = \mathcal{Q} \cup \{\hat{Q}\}$ where $\mathcal{Q} = \bigcup_{v\in[n], j\in[K]}\{Q_{v,j}\}$ (with duplicates removed) and $\hat{Q} := \operatorname{cl}\left([0, K]^n \setminus \bigcup_{Q\in\mathcal{Q}} Q\right)$, which is the closure of the part of $[0, K]^n$ that is not covered by $\mathcal{Q}$. We refer to $\hat{Q}$ as the leftover region of $[0, K]^n$. The cardinality of $\mathcal{Q}'$ is $n(K-1)+2$ because $Q_{1,K} = \cdots = Q_{n,K}$. Figure 2a graphically depicts $\mathcal{Q}'$ when $n = 3$ and $K = 4$.*



(a) (b) (c)

Fig. 2: Example partition (a), regular refinement (b), and non-regular refinement (c) when $n = 3$ and $K = 4$.

It is clear that any RHP that refines the partition $\mathcal{Q}'$ of Example 2 requires at least K^n hyper-rectangles. We show in Proposition 3.1 that there exists a non-regular hyper-rectangular refinement $\{Q'_i\}_{i\in[L']}$ of $\mathcal{Q}'$ with $L' = \mathcal{O}(n^2 + nK)$.

Proposition 3.1. *For any positive integer parameters n and K, there is a (non-regular) HP with cardinality $\frac{n(n-3)}{2}+nK+1$ that refines the partition $\mathcal{Q}'$ of Example 2.*

Proof. We prove the statement by defining a partition $\mathcal{R}$ of the leftover region $\hat{Q}$ whose cardinality is $\binom{n}{2}$. Let $\mathcal{R} = \{R_{v_1,v_2}\}_{v_1<v_2\in[n]}$, where

$$R_{v_1,v_2} = \{\boldsymbol{z} \in [0,K]^n \mid z_{v_1}, z_{v_2} \le K-1, z_v \ge K-1, \forall v \in [v_2-1] : v \ne v_1\}, \tag{3.1}$$

for all $v_1, v_2, \in [n]$ such that $v_1 < v_2$.

Clearly, every R_{v_1,v_2} in $\mathcal{R}$ is a hyper-rectangle. We show that $\mathcal{R}$ is a partition of $\hat{Q}$, *i.e.*, (i) $\bigcup_{R\in\mathcal{R}} R = \hat{Q}$ and (ii) the interior of $R_{a,b}$ and $R_{c,d}$ in $\mathcal{R}$ are disjoint if $a \ne c$ or $b \ne d$. We first prove (i). We can write $\hat{Q} = \{\boldsymbol{z} \in [0,K]^n \mid |\{v \in [n] \mid z_v \le K-1\}| \ge 2\}$. It follows that $\bigcup_{R\in\mathcal{R}} R \subseteq \hat{Q}$. Also, $\hat{Q} \subseteq \bigcup_{R\in\mathcal{R}} R$ because for a point $\boldsymbol{z} \in \hat{Q}$, $\boldsymbol{z} \in R_{v_1,v_2}$ where v_1, v_2 are the first two indices for which $z_v \le K-1$. We next prove (ii). Consider $R_{a,b}$ and $R_{c,d}$ with $a < b$ and $c < d$. We consider two cases. First, suppose that $a \ne c$. Without loss of generality, assume that $a < c$. Then, hyperplane $z_a = K-1$ separates $R_{a,b}$ and $R_{c,d}$ as $R_{a,b}$ satisfies $z_a \le K-1$ and $R_{c,d}$ satisfies $z_a \ge K-1$. Second, suppose that $a = c$. Without loss of generality, assume that $b < d$. Then, hyperplane $z_b = K-1$ separates $R_{a,b}$ and $R_{c,d}$ as $R_{a,b}$ satisfies $z_b \le K-1$ and $R_{c,d}$ satisfies $z_b \ge K-1$. Therefore, the interiors of distinct sets $R_{a,b}$ and $R_{c,d}$ in $\mathcal{R}$ are disjoint. In conclusion, $\mathcal{Q} \cup \mathcal{R}$ forms an HP of $[0,K]^n$ with cardinality $|\mathcal{Q}| + |\mathcal{R}| = (n(K-1)+1) + \frac{n(n-1)}{2} = \frac{n(n-3)}{2} + nK + 1$. ∎

Remark 3.2. The non-regular HP in the proof of Proposition 3.1 can be obtained as leaves of a branch-and-bound tree/decision tree. In particular, consider branching so that $z_{v_1} \le K-1$ and $z_{v_1} \ge K-1$. Let us consider the branch $z_{v_1} \le K-1$. If we branch on another variable, say z_{v_2} so that $z_{v_2} \le K-1$, then the node belongs to the region $\hat{Q}$ which was left-over in $[0,K]^n$. On the other hand, if all the remaining variables are such that $z_{v_i} \ge K-1$, for $i > 1$ then the particular value of z_{v_1} can be found with $K-1$ leaves in a straightforward way. The branch where $z_{v_1} \ge K-1$ reduces to a similar problem with one less variable. Therefore, if $L(n)$ denotes the number of leaves in the branch-and-bound tree with n variables, the linear recurrence relationship $L(n) = n + K - 2 + L(n-1)$ must hold with $L(1) = K$. Solving this recurrence, we obtain that $L(n) = \frac{n(n-3)}{2} + nK + 1$, which matches the result of Proposition 3.1. This result has the consequence, which may be of independent interest, that regular-partitioning can require exponentially many nodes compared to a branch-and-bound tree or, equivalently, a decision tree representation.

3.2. An MILP formulation Over the Union of Polytopes. In this section, we provide an MILP formulation for a PPR of a single function $g(\boldsymbol{z})$ over a polyhedral partition of $\mathcal{Z}$. We remark that $g(\boldsymbol{z})$ does not need to be multilinear and the partition does not need to be hyper-rectangular for the results in this section to be used. Let $\{Q_i\}_{i\in[L]}$ be a polyhedral partition of $\mathcal{Z}$ and let $\{\bar{Q}_i\}_{i\in[L]}$ be a PPR of h over $\{Q_i\}_{i\in[L]}$. We next provide an MILP formulation for $(\boldsymbol{z}, w) \in \bigcup_{i\in[L]} \bar{Q}_i$.

We use four types of decision variables: $\boldsymbol{z}$, w, $\boldsymbol{y}$, and $\boldsymbol{\lambda}$. Variables $(\boldsymbol{z}, w) \in \mathbb{R}^{n+1}$ represent points in $\bigcup_{i\in[L]} \bar{Q}_i$. Variable $\lambda_{(\bar{\boldsymbol{z}},\bar{w})}$ represents the convex combination weight for $(\bar{\boldsymbol{z}}, \bar{w})$ for all $(\bar{\boldsymbol{z}}, \bar{w}) \in \bar{\mathcal{Z}} := \bigcup_{i\in[L]} \operatorname{vert} \bar{Q}_i$. We denote by $n_\lambda = |\bar{\mathcal{Z}}|$ the number of λ variables. Variables $\boldsymbol{y} = \{y_i\}_{i\in[L]} \in \Delta^L_{0,1}$ form a unit vector whose single index i taking the value 1 carries the information that $\boldsymbol{z} \in Q_i$. When $\boldsymbol{y} = e_i$, we will refer to Q_i as the *active polytope*.

Variables $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ satisfy the following disjunction:

$$(\boldsymbol{\lambda}, \boldsymbol{y}) \in \bigvee_{i\in[L]} \left\{ (\boldsymbol{\lambda}, \boldsymbol{y}) \in \Delta^{n_{\boldsymbol{\lambda}}} \times \Delta^L_{0,1} \;\middle|\; \begin{array}{l} y_i = 1, \\ \lambda_{(\bar{\boldsymbol{z}},\bar{w})} = 0, \forall (\bar{\boldsymbol{z}}, \bar{w}) \in \bar{\mathcal{Z}} \setminus \operatorname{vert} \bar{Q}_i \end{array} \right\}. \tag{3.2}$$

To derive linear constraints for (3.2), we define bipartite graph $G = (U, V, E)$ where $U = \bar{\mathcal{Z}}$, $V = [L]$, and there exists an edge between $(\bar{\boldsymbol{z}}, \bar{w}) \in U$ and $i \in V$ if $(\bar{\boldsymbol{z}}, \bar{w}) \in \bar{Q}_i$. We denote by $N_G(u) = \{v \in V \mid (u, v) \in E\}$ for $u \in U$ the set of the neighbors of node u in G. Similar to our derivations in Section 4.1 of [16], a convex hull description of (3.2) can be derived using Hoffman's circulation theorem [14]. This description is comprised of the following constraints:

$$\sum_{u\in U: N_G(u)\subseteq S} \lambda_u \le \sum_{i\in S} y_v, \qquad \forall S \subsetneq V : S \ne \emptyset. \tag{3.3}$$

We next present an MILP formulation for $(\boldsymbol{z}, w) \in \bigcup_{i\in[L]} \bar{Q}_i$. We use (3.4b)–(3.4f) instead of (3.3) to describe the convex hull of (3.2) using additional variables $\{h_e\}_{e\in E}$ since the number of variables and constraints this formulation requires is polynomial in the number of vertices and edges of G:

$$\begin{pmatrix} \boldsymbol{z} \\ w \end{pmatrix} = \sum_{(\bar{\boldsymbol{z}},\bar{w})\in\bar{\mathcal{Z}}} \begin{pmatrix} \bar{\boldsymbol{z}} \\ \bar{w} \end{pmatrix} \lambda_{(\bar{\boldsymbol{z}},\bar{w})}, \tag{3.4a}$$

$$\sum_{e=(u',v')\in E: u'=u} h_e = \lambda_u \qquad \forall u \in U, \tag{3.4b}$$

$$\sum_{e=(u',v')\in E: v'=v} h_e = y_i \qquad \forall v \in V, \tag{3.4c}$$

$$\boldsymbol{\lambda} \in \Delta^{n_{\lambda}}, \tag{3.4d}$$

$$h_e \ge 0, \qquad \forall e \in E, \tag{3.4e}$$

$$\boldsymbol{y} \in \Delta^L_{0,1}. \tag{3.4f}$$

Formulation (3.4) is ideal because $\boldsymbol{z}$ and w are dependent on $\boldsymbol{\lambda}$ and (3.4b)–(3.4f) describe the convex hull of the system of $\boldsymbol{\lambda}$ and $\boldsymbol{y}$.

We believe that formulation (3.4) has advantages over the formulations that can be obtained for this set using results in the literature. Consider for instance the formulations for piecewise linear functions over the union of polyhedra presented in [31]. Two of these formulations utilize convex combination variables, which are similar to variable $\boldsymbol{\lambda}$ in our formulation.

The first formulation is called the *aggregated convex combination* model, which uses the variables $(\boldsymbol{x}, w, \boldsymbol{\lambda}, \boldsymbol{y})$ of (3.4). It is a sharp formulation, where a formulation for $(\boldsymbol{z}, w) \in \bigcup_{i\in[L]} \bar{Q}_i$ is said to be *sharp* if the projection of its feasible set over the space of $(\boldsymbol{z}, w)$ variables is $\operatorname{conv}(\bigcup_{i\in[L]} \bar{Q}_i)$. Clearly, every ideal formulation is sharp but the opposite direction need not hold. It follows that, when constructing an MIP model using multiple PPRs for different components, using sharp formulations often results in a weaker LP relaxation than using ideal formulations.

The second formulation is called the *disaggregated convex combination* model. It is an ideal formulation that introduces separate λ variables for the same $\bar{\boldsymbol{z}}$ if multiple

$\bar{Q}_i$ share $\bar{\boldsymbol{z}}$ as their vertex. In the context of our paper, especially when $\{Q_i\}_{i\in[L]}$ is an HP of $S \subseteq \mathbb{R}^n$, it introduces up to 2^n variables for the same vertex, which can be much larger than the number of variables we use.

Hence, formulation (3.4) has significant potential advantages for the solution of (1.2) that we develop in the next section.

3.3. A MILP Formulation Over Non-Regular HPs. In this section, we introduce a novel MILP formulation for (1.2) that builds on the advantageous characteristics of each of the formulations presented in section 2.1 and section 3.2 while overcoming some of their disadvantages.

Consider first (IPPR1) in section 2.1. This formulation assumes that HPs are regular. It uses positioning variables $\boldsymbol{x}$ that have the advantage of modeling the geometry of the problem. In the literature, variables $t_{v,j} = \sum_{j'\leq j} x_{v,j'}$, referred to as *incremental variables*, are often used instead of $\boldsymbol{x}$. Incremental variables tend to lead to better branching decisions, informed by the geometry of the problem, because $t_{v,j}$ takes value 1 (resp. 0) only if $z_v \leq d_{v,j+1}$ (resp. $z_v \geq d_{v,j+1}$). We could seek to take advantage of these benefits by utilizing a RHP that refines the given non-regular HP. We have demonstrated, however, in Example 2 that such construction might require K^n hyper-rectangles while the given non-regular HP might only be composed of $\frac{n(n-3)}{2} + nK + 1$ hyper-rectangles.

Consider second formulation (3.4) in section 3.2. In situations such as those illustrated in Example 2, this formulation can be used to avoid generating exponentially many hyper-rectangles. However, it is not clear how to connect different partitions $\{Q_{t,i}\}_{i\in[L_t]}$ as the formulation does not contain the positioning variables $\boldsymbol{x}$ introduced for RHPs in section 2.1.

We propose next an MILP formulation for (1.2) that combines the advantages of the formulations described in previous sections by using both $\boldsymbol{x}$ and $\boldsymbol{y}$ variables. This MILP formulation, which we refer to as (IPPR2), applies when $\{Q_{t,i}\}_{i\in[L_t]}$ is an HP of $\mathcal{Z}_{J_t}$ for all $t \in [n_Q]$.

We use six types of decision variables, $\boldsymbol{z}$, $\boldsymbol{w}$, $\boldsymbol{\lambda}$, $\boldsymbol{h}$, $\boldsymbol{y}$, and $\boldsymbol{x}$. Variables $\boldsymbol{z}$ and $\boldsymbol{w}$ are the same variables used in (1.1). For $t \in [n_Q]$, we denote by $\bar{\mathcal{Z}}_t = \bigcup_{i\in[L_t]} \operatorname{vert} Q_{t,i}$ the set of all vertices used in a convex combination that expresses the PPRs of a multilinear function defined over the t^{th} HP. Variable $\lambda^t_{\bar{\boldsymbol{z}}}$ indicates the convex combination weight for vertex $\bar{\boldsymbol{z}}$ in the space of $\boldsymbol{z}_{J_t}$ for all $t \in [n_Q]$ and for all $\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_t$. For $j \in [n_Q]$, binary variable $y_{t,i}$ is 1 if and only if $Q_{t,i}$ is active among $\{Q_{t,i}\}_{i\in[L_t]}$. Similar to section 3.2, for each $t \in [n_Q]$, we construct bipartite graph $G_t = (U_t, V_t, E_t)$ where $U_t = \bar{\mathcal{Z}}_t$, $V_t = [L_t]$, and $E_t = \{(\bar{\boldsymbol{z}}, i) \in U_t \times V_t \,|\, \bar{\boldsymbol{z}} \in \operatorname{vert} Q_{t,i}\}$. Then, variable $\boldsymbol{h} = \{h_{t,e}\}_{\forall t\in[n_Q],\forall e\in E_t}$ can be used to relate $\boldsymbol{\lambda}$ and $\boldsymbol{y}$. Finally, binary variable $x_{v,k}$ indicates the k^{th} interval on the z_v-axis for all $v \in [n]$ and for all $k \in [D_v]$, where the discretization points $\{d_{v,k}\}_{k\in[D_v]}$ are collected from all HPs $\{Q_{t,i}\}_{i\in[L_t]}$ for all $t \in [n_Q]$.

We relate $\boldsymbol{x}$ and $\boldsymbol{y}_t = \{y_{t,i}\}_{i\in[L_t]}$ which independently indicate the active hyper-rectangle among $\{Q_{t,i}\}_{i\in[L_t]}$ for a fixed $t \in [n_Q]$. For a hyper-rectangle $Q_{t,i}$, we define $k_1(t,i,v) = \min\{k \in [D_v - 1] \,|\, \delta_{v,k} \in \operatorname{proj}_{z_v} Q_{t,i}\}$ and $k_2(t,i,v) = \max\{k \in [D_v - 1] \,|\, \delta_{v,k} \in \operatorname{proj}_{z_v} Q_{t,i}\}$ to indicate the leftmost and the rightmost intervals on the z_v-axis that $Q_{t,i}$ overlaps, respectively. Variables $\boldsymbol{x}$ and $\boldsymbol{y}_t$ satisfy the following multilinear constraint:

(3.5)

$$(\boldsymbol{x}, \boldsymbol{y}_t) \in \left\{ (\boldsymbol{x}, \boldsymbol{y}_t) \in \prod_{v\in[n]} \Delta_{0,1}^{D_v-1} \times \Delta^{L_t} \;\middle|\; y_{t,i} = \prod_{v\in[n]} \sum_{k=k_1(t,i,v)}^{k_2(t,i,v)} x_{v,k}, \forall i \in [L_t] \right\}.$$

Such relationship between $\boldsymbol{x}$ and $\boldsymbol{y}_t$ is a *facial decomposition of the Cartesian product of simplices.* An explicit convex hull description of (3.5) is provided in Theorem 3 in [16]; this description is in fact the system of (3.6g), (3.6j), and (3.6k) with fixed t. We thus obtain the following formulation (IPPR2):

$$
\begin{aligned}
&\text{(3.6a)} \quad && \max \quad \boldsymbol{c}_{\boldsymbol{z}}^{\intercal}\boldsymbol{z} + \boldsymbol{c}_{\boldsymbol{w}}^{\intercal}\boldsymbol{w} && \\
&\text{(3.6b)} && \text{s.t.} \quad A_{\boldsymbol{z}}\boldsymbol{z} + A_{\boldsymbol{w}}\boldsymbol{w} \leq \boldsymbol{b}, && \\
&\text{(3.6c)} && \boldsymbol{z}_{J_t} = \sum_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_t} \bar{\boldsymbol{z}} \lambda_{\bar{z}}^t, && \forall t \in [n_Q], \\
&\text{(3.6d)} && w_j = \sum_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_{\sigma(j)}} f_j(\bar{\boldsymbol{z}}) \lambda_{\bar{z}}^{\sigma(j)}, && \forall j \in [n_w], \\
&\text{(3.6e)} && \sum_{e=(u',v') \in E_t : u'=u} h_{t,e} = \lambda_u^t, && \forall t \in [n_Q], \forall u \in U_t, \\
&\text{(3.6f)} && \sum_{e=(u',v') \in E_t : v'=v} h_{t,e} = y_{t,i}, && \forall t \in [n_Q], \forall v \in V_t, \\
&\text{(3.6g)} && \sum_{i \in [L_t] : k_1 \leq k_1(t,i,v), k_2(t,i,v) \leq k_2} y_{t,i} \leq \sum_{k=k_1}^{k_2} x_{v,k}, && \forall t \in [n_Q], \forall v \in [n], \\
& && && \forall k_1 \leq k_2 \in [D_v - 1], \\
&\text{(3.6h)} && \boldsymbol{\lambda^t} = \{\lambda_{\bar{\boldsymbol{z}}}^t\}_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_t} \in \Delta^{|\bar{\mathcal{Z}}_t|}, && \forall t \in [n_Q], \\
&\text{(3.6i)} && h_{t,e} \geq 0, && \forall t \in [n_Q], \forall e \in E_t, \\
&\text{(3.6j)} && \boldsymbol{y_t} \in \Delta^{L_t}, && \forall t \in [n_Q], \\
&\text{(3.6k)} && \boldsymbol{x_v} \in \Delta_{0,1}^{D_v - 1}, && \forall v \in [n].
\end{aligned}
$$

THEOREM 3.3. *(IPPR2) is a locally ideal formulation for* (1.2) *and its size is polynomial in n, n_w, n_A, and the maximum number of vertices in an HP, i.e.,* $\max_{t \in [n_Q]} |\mathcal{Z}_t|$.

Proof. We first show that the size of (IPPR2) is polynomial in n, n_w, n_A, and $\max_{t \in [n_Q]} |\mathcal{Z}_t|$. We also use the number of HP (n_Q) and the maximum number of elements in an HP ($\max_{t \in [n_Q]} L_t$) when deriving the bounds because $n_Q \leq n_w$ and $\max_{t \in [n_Q]} L_t \leq \max_{t \in [n_Q]} |\mathcal{Z}_t|$. The numbers of $\boldsymbol{z}$ and $\boldsymbol{w}$ variables are n and n_w, respectively. The number of $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ variables is $\sum_{t \in [n_Q]} |\mathcal{Z}_t|$ and $\sum_{t \in [n_Q]} L_t$, respectively, which are bounded above by $n_Q \max_{t \in [n_Q]} |\mathcal{Z}_t|$ and $n_Q \max_{t \in [n_Q]} L_t$, respectively. The number of $\boldsymbol{h}$ variables is $\sum_{t \in [n_Q]} |E_t|$, which is bounded above by $n_Q \left(\max_{t \in [n_Q]} |L_t|\right) \left(\max_{t \in [n_Q]} |\mathcal{Z}_t|\right)$. The number of $\boldsymbol{x}$ variables is $\sum_{v \in [n]} (D_v - 1)$, which is bounded above by $2n \max_{t \in [n_Q]} |L_t|$. The number of constraints is polynomial in n, n_w, n_A, and $\max_{t \in [n_Q]} |\mathcal{Z}_t|$ because it holds that (i) $D_v \leq \max_{t \in [n_Q]} 2|L_t|$, for all $v \in [n]$ and (ii) $|U_t| \leq \max_{t' \in [n_Q]} |\mathcal{Z}_{t'}|$, $|V_t| \leq \max_{t' \in [n_Q]} L_{t'}$, and $|E_t| \leq |U_t||V_t|$ for all $t \in [n_Q]$. Therefore, the total number of variables and constraints of (IPPR2) is polynomial in n, n_w, n_A, and $\max_{t \in [n_Q]} |\mathcal{Z}_t|$.

We next show that (IPPR2) is locally ideal. Suppose $n_Q = 1$. Consider the set S_1 in the space of variables $(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{h}, \boldsymbol{y})$ obtained by retaining all of the constraints containing only these variables. Let S_2 be the set in the space of variables $(\boldsymbol{y}, \boldsymbol{x})$

obtained by retaining all of the constraints containing only these variables. Observe that every variable/constraint belongs to at least one of S_1 and S_2. Therefore, the feasible set S of (3.6) can be written as

$$S = \{(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{h}, \boldsymbol{y}, \boldsymbol{x}) \mid (\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{h}, \boldsymbol{y}) \in S_1, (\boldsymbol{y}, \boldsymbol{x}) \in S_2\}.$$

Set S_1 is integral because (3.4) is ideal. Set S_2 is integral by Theorem 3 in [16]. Then, S is also integral; see [27] or [16], which states that S is integral if both S_1 and S_2 are integral and the common variable $\boldsymbol{y}$ forms a simplex. Therefore, (IPPR2) is ideal when $n_Q = 1$. ∎

A distinct advantage of our approach is that it allows the incorporation of geometrical information into models defined over non-regular partitions of their domain, without requiring that the partition be first subdivided into one that is regular. The formulations so-produced therefore have the advantage of typically requiring fewer variables without compromising on their convex hull properties. Intuitively, they combine the advantages of previously proposed approaches. In particular, the presence of positioning variables $\boldsymbol{x}$ might prove helpful in guiding branching decisions. In this respect, one could straightforwardly make use of incremental t variables in our formulations, without compromising on their strength, as variables $x_{v,j}$ are related to these variables via the linear and invertible transformation, $x_{v,j} = t_{v,j} - t_{v,j-1}$, where $t_{v,0} = 0$. Formulation (IPPR2) is, to the best of the authors' knowledge, the first formulation modeling (1.2) over general HPs that is locally ideal and whose size is polynomial in n, n_w, n_A, and $\max_{t\in[n_Q]} |\mathcal{Z}_t|$.

3.4. Extension of Linking Constraints. In this section, we extend the applicability of linking constraints (2.13) from (IPPR1) to the case of (IPPR2). We first introduce constraints linking $\boldsymbol{\lambda}^{t_1}$ and $\boldsymbol{\lambda}^{t_2}$ variables for $t_1 \neq t_2 \in [n_Q]$. Let $\mathcal{S} = \{S \subseteq [n] \setminus \{\emptyset\} \mid \exists t_1, t_2 \in [n_Q] : S \subseteq J_{t_1} \cap J_{t_2}\}$.

PROPOSITION 3.4. *Consider formulation (IPPR2), i.e., all the variables and constraints in (3.6), together with additional variable μ_S for $S \in \mathcal{S}$ and the following constraints*

$$\mu_S = \sum_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_t} \left(\prod_{v \in S} \bar{\boldsymbol{z}}_{(v)} \right) \lambda^t_{\bar{\boldsymbol{z}}}, \qquad \forall (S,t) \in \mathcal{S} \times [n_Q] : S \subseteq J_t. \tag{3.7}$$

This formulation, which we refer to as (PPR2), is a relaxation of (1.1), i.e., the projection in the space of $(\boldsymbol{z}, \boldsymbol{w})$ of the feasible set of (PPR2) contains the feasible set of (1.1).

Proof. Consider a feasible solution of $(\bar{\boldsymbol{z}}, \bar{\boldsymbol{w}})$ of (1.1). Construct $\bar{\boldsymbol{\mu}}$ by assigning $\bar{\mu}_S = \prod_{v\in S} \bar{z}_{(v)}$. We prove the statement by constructing $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{h}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$ such that (i) $(\bar{\boldsymbol{z}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{h}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$ is feasible to (IPPR2) and (ii) $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})$ satisfies (3.7).

For each $v \in [n]$, choose $k_v \in [D_v]$ such that $\bar{z}_v \in \delta_{v,k_v}$. We construct $\bar{\boldsymbol{x}}$ by assigning $\bar{x}_{v,k_v} = 1$ and $\bar{x}_{v,k} = 0$ for all $k \in [D_v] \setminus \{k_v\}$ for each $v \in [n]$. We then construct $\bar{\boldsymbol{y}}$ using (3.5). Let i_t be the index of the active hyper-rectangle for each $t \in [n_Q]$. According to the derivation of the convex hull description of (3.2) in section 3.2, if $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{y}})$ satisfies (3.2), then we can always construct $\bar{\boldsymbol{h}}$ such that $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{h}}, \bar{\boldsymbol{y}})$ satisfies all constraints that $\boldsymbol{h}$ appears in, *i.e.*, (3.6e), (3.6f), and (3.6i). We fix $\lambda^t_{\bar{\boldsymbol{z}}} = 0$ for $\bar{\boldsymbol{z}} \notin \operatorname{vert} Q_{t,i_t}$ by (3.2). Then, it is sufficient to construct $\{\bar{\lambda}^t_{\bar{\boldsymbol{z}}}\}_{\bar{\boldsymbol{z}} \in \operatorname{vert} Q_{t,i_t}} \in \Delta^{2^{|J_t|}}$ for all $t \in [n_Q]$ that satisfies (3.6c), (3.6d), and (3.7) given $(\bar{\boldsymbol{z}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{\mu}})$. For $t \in [n_Q]$, we construct $\{\bar{\lambda}^t_{\bar{\boldsymbol{z}}}\}_{\bar{\boldsymbol{z}} \in \operatorname{vert} Q_{t,i_t}}$ using the following procedure from (a) to (c): (a) let $\ell(t, i_t, v) =$

$\min_{\bar{\boldsymbol{z}}\in Q_{t,i_t}} \bar{\boldsymbol{z}}_{(v)}$ and $u(t,i_t,v) = \max_{\bar{\boldsymbol{z}}\in Q_{t,i_t}} \bar{\boldsymbol{z}}_{(v)}$, (b) let $\lambda^t_{(v)} = \frac{\bar{z}_v - \ell(t,i_t,v)}{u(t,i_t,v)-\ell(t,i_t,v)}$ for all $v \in J_t$, (c) assign $\lambda^t_{\bar{\boldsymbol{z}}} = \prod_{v\in J_t:\bar{\boldsymbol{z}}_{(v)}=u(t,i_t,v)} \lambda^t_{(v)} \prod_{v\in J_t:\bar{\boldsymbol{z}}_{(v)}=\ell(t,i_t,v)}(1-\lambda^t_{(v)})$ for all $\bar{\boldsymbol{z}} \in \operatorname{vert} Q_{t,i_t}$. We next prove Lemma 3.5 that states that for $t \in [n_Q]$, if $\boldsymbol{\lambda}^t$ is constructed using the above procedure, then the value of its convex combination expression of a multilinear function is equal to the function value.

LEMMA 3.5. *Given $\boldsymbol{\ell}, \boldsymbol{u} \in \mathbb{R}^n$, consider a hyper-rectangle $Q = \{\boldsymbol{z} \in \mathbb{R}^n \mid \ell_v \le z_v \le u_v, \forall v \in [n]\}$. Given $\boldsymbol{z} \in \mathbb{R}^n$, define $\lambda_v = \frac{z_v - l_v}{u_v - l_v}$ for $v \in [n]$. Define $\boldsymbol{\lambda} = \{\lambda_{\bar{\boldsymbol{z}}}\}_{\bar{\boldsymbol{z}}\in\operatorname{vert} Q}$ as*

$$\lambda_{\bar{\boldsymbol{z}}} = \prod_{v\in[n]:\bar{z}_v=u_v} \lambda_v \prod_{v\in[n]:\bar{z}_v=\ell_v}(1-\lambda_v), \qquad \forall \bar{\boldsymbol{z}} \in \operatorname{vert} Q. \tag{3.8}$$

Then, for any multilinear function $f(\boldsymbol{z})$, it holds that

$$f(\boldsymbol{z}) = \sum_{\bar{\boldsymbol{z}}\in\operatorname{vert} Q} f(\bar{\boldsymbol{z}})\lambda_{\bar{\boldsymbol{z}}}. \tag{3.9}$$

Proof. It is sufficient to show that (3.9) holds when $f(\boldsymbol{z}) = \prod_{v\in I} z_v \prod_{v\in[n]\setminus I}(1 - z_v)$ for all $I \subseteq [n]$ because $\{\prod_{v\in I} z_v \prod_{v\in[n]\setminus I}(1-z_v)\}_{I\subseteq[n]}$ is a basis of multilinear functions in the space of $\boldsymbol{z}$. By applying an affine transformation if necessary, we may assume that $\ell_v = 0$ and $u_v = 1$ for all $v \in [n]$. Consider any $I \subseteq [n]$. We denote by $\bar{\boldsymbol{z}}^*$ the vector in $\mathbb{R}^n$ such that $\bar{\boldsymbol{z}}^*_v = 1$ if $v \in I$ and $\bar{\boldsymbol{z}}^*_v = 0$ otherwise. It holds that

$$\sum_{\bar{\boldsymbol{z}}\in\operatorname{vert} Q} f(\bar{\boldsymbol{z}})\lambda_{\bar{\boldsymbol{z}}} = f(\bar{\boldsymbol{z}}^*)\lambda_{\bar{\boldsymbol{z}}^*} = \prod_{v\in I}\lambda_v \prod_{v\in[n]\setminus I}(1-\lambda_v) = \prod_{v\in I} z_v \prod_{v\in[n]\setminus I}(1-z_v) = f(\boldsymbol{z}), \tag{3.10}$$

where the first equality is obtained by removing zero-coefficient terms, the second equality is derived because $f(\bar{\boldsymbol{z}}^*) = 1$ and (3.8), the third equality holds by $\lambda_v = \frac{z_v - l_v}{u_v - l_v} = \frac{z_v - 0}{1 - 0} = z_v$ for all $v \in [n]$, and the last equality holds by the assumption. Hence, the lemma is proven. ∎

By Lemma 3.5, $\bar{\boldsymbol{\lambda}}$ constructed by the procedure (a)–(c) together with $\bar{\boldsymbol{z}}$, $\bar{\boldsymbol{w}}$, and $\bar{\boldsymbol{\mu}}$ satisfy (3.6c), (3.6d), and (3.7). It is because the right-hand-sides of (3.6c), (3.6d), and (3.7) are the convex combination expressions of a linear/multilinear function of $\boldsymbol{z}$ using $\boldsymbol{\lambda}^t$ for some $t \in [n_Q]$ and $\bar{\boldsymbol{z}}$, $\bar{\boldsymbol{w}}$, and $\bar{\boldsymbol{\mu}}$ are their function values. Therefore, $(\bar{\boldsymbol{z}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{h}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$ is feasible to (IPPR2) and $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})$ satisfies (3.7). ∎

Similar to (2.13), the rhs of (3.7) for $(S,t) \in \mathcal{S} \times [n_Q]$ with $S \subseteq J_t$ can be interpreted as the convex combination using $\boldsymbol{\lambda}^t$ variable that expresses $\prod_{v\in S} z_v$. Clearly, (3.7) is a generalization of (2.13).

If two partitions are the same, it is trivial to extend the linking constraints (2.10) so that the multipliers associated with the corresponding extreme points from both partitions are equated with one another. This extension applies when a common refinement of subspaces of partitions is easy to construct. Unfortunately, if each partition has a different set of polynomially many non-regular elements on a common set of variables and the number of partitions is bounded by $\binom{n}{2}$, it was shown in Theorem 4 of [16] that a common refinement may require exponentially many elements. Moreover, [16] discussed settings where a polynomially-sized common refinement can be constructed. It follows that in the latter cases, it is easy to generalize the linking constraints. However, if a common refinement requires exponentially many elements, it is not straightforward to generalize the linking constraints while retaining their polynomial size. This issue does not arise with RHPs with the same discretization points since they automatically share a common refinement on each subspace.

3.5. Computational Experiments with Regular and Non-Regular HPs. In this section, we perform experiments to demonstrate that non-regular HPs have computational advantages compared to RHPs. We consider variants of *tree ensemble optimization (TEO)* problems [21]. A tree ensemble model is a collection of decision trees. Traditionally, decision trees model a piecewise constant function over a (usually non-regular) HP. Then, TEO seeks to find values for the input variables of a given tree ensemble model so as to minimize/maximize the prediction value. TEO has been used to find the best combination of compounds to design new drugs [21] and to find optimal assortments in marketing that maximize profit [2].

In this experiment, we consider linear regression trees instead of classical decision trees as the elements of the tree ensemble model. A *linear regression tree* [26, 4] associates a linear model with each leaf. Given an input value, the prediction from the ensemble is computed by averaging the predictions from each tree. The prediction from a tree is obtained by using the linear model that is associated with the leaf to which the input value belongs. We remark that linear regression trees produce piecewise linear functions that generalize the type of functions obtained using classical decision trees. We recall that our formulations result in a valid relaxation only if the projection of the extreme points of the graph of the functions over an HP $\{Q_{t,i}\}_{i\in[L_t]}$ is contained in the vertices $\bar{\mathcal{Z}}_t$ of the HP. In fact, it follows that tree ensembles of multilinear decision trees introduced in [16] can also be relaxed using our techniques. It also follows easily that the functions arising from linear regression trees satisfy this property. Moreover, with piecewise linearity, our formulations are exact and do not require partitioning over continuous variables in a branch-and-bound algorithm.

We now describe how our formulation takes advantage of the partition structure inherent to the linear regression tree. We denote by n the number of input variables of the given tree ensemble model and suppose that the domain of the input variable $\boldsymbol{z}$ is $\mathcal{Z}$. We denote by n_w the number of trees in the ensemble and by L_j the number of leaves in the j^{th} tree for $j \in [n_w]$. The set of leaves of a decision tree corresponds to an HP of its domain because each nonleaf node of the tree divides the domain using a hyperplane $z_v = a$ for some $a \in \mathbb{R}$. We denote by $Q_{j,i}$ and $f_{j,i}(\boldsymbol{z})$ the hyper-rectangle and the linear function corresponding to the i^{th} leaf in the j^{th} tree, respectively, for $j \in [n_w]$ and for $i \in [L_j]$. Using this notation, we formulate the problem as

$$\max \quad \frac{1}{n_w}\sum_{j\in[n_w]} w_j \tag{3.11a}$$

$$\text{s.t.} \quad \begin{pmatrix} \boldsymbol{z} \\ w_j \end{pmatrix} \in \bigcup_{i\in[L_j]} \left\{ \begin{pmatrix} \boldsymbol{z} \\ f_{j,i}(\boldsymbol{z}) \end{pmatrix} \;\middle|\; \boldsymbol{z} \in Q_{j,i} \right\}, \qquad \forall j \in [n_w], \tag{3.11b}$$

$$\boldsymbol{z} \in \mathcal{Z}. \tag{3.11c}$$

We assume, as is typical for TEO problems, that when the value of $\boldsymbol{z}$ lies on the boundary of multiple hyper-rectangles, model (3.11) is free to select w_j using any of the corresponding linear functions. Constraint (3.11b) can be written as $(\boldsymbol{z}, w_j) \in \bigcup_{i\in[L_j]} \operatorname{conv}\{(\boldsymbol{z}, f_{j,i}(\boldsymbol{z}))\}_{\boldsymbol{z}\in \operatorname{vert} Q_{j,i}}$ for all $j \in [m]$ because the functions $f_{j,i}(\boldsymbol{z})$ are linear over each $Q_{j,i}$. Then, (3.11) takes the form of a PPR of an optimization problem over HPs.

The partitions used in the above problem are typically not regular. We could build an alternative formulation by constructing a RHP $\{Q'_{j,i}\}_{i\in[L'_j]}$ that refines $\{Q_{j,i}\}_{i\in L_j}$

Table 3: Solution times and number of hyper-rectangles in regular/non-regular HPs for TEO instances with n input variables and n_w trees with maximum depth D.

Data set	n	D	n_w	# of hyper-rectangles		Solution times	
				Non-Regular	Regular	Non-Regular	Regular
diabetes	10	2	5	20	38	1.7	0.8
diabetes	10	2	10	40	76	9.9	12.3
diabetes	10	2	15	59	112	12.4	64.6
diabetes	10	2	20	77	140	21.3	120.8
diabetes	10	3	5	29	120	2.9	44.4
diabetes	10	3	10	59	224	15.4	23.3
diabetes	10	3	15	88	338	28.1	212.0
diabetes	10	3	20	119	498	35.5	301.7
diabetes	10	4	5	35	322	10.4	384.7
diabetes	10	4	10	72	666	31.2	963.7
diabetes	10	4	15	106	894	63.0	1962.9
diabetes	10	4	20	144	1176	106.2	3600.0
house price	8	2	5	20	40	0.2	0.8
house price	8	2	10	40	80	0.6	1.0
house price	8	2	15	60	120	2.4	6.8
house price	8	2	20	80	158	4.6	13.6
house price	8	3	5	27	72	0.4	3.5
house price	8	3	10	56	176	2.2	21.5
house price	8	3	15	86	296	1.2	11.0
house price	8	3	20	114	380	10.3	65.1
house price	8	4	10	69	432	1.8	55.0
house price	8	4	15	105	664	5.4	139.8
house price	8	4	20	143	890	10.6	225.3

for all $j \in [n_w]$, where L'_j is a positive integer. Specifically, for $j \in [n_w]$, $\{Q'_{j,i}\}_{i\in[L'_j]}$ is constructed using the discretization points that appear in the j^{th} decision tree. The linear function $f'_{j,i'}(\boldsymbol{z})$ associated with $Q'_{j,i'}$ is defined as $f_{j,i}$ when $Q'_{j,i'} \subseteq Q_{j,i}$. It is clear that the TEO problem constructed using $\{Q'_{j,i}\}_{i\in[L'_j]}$ and $\{f'_{j,i'}(\boldsymbol{z})\}_{i\in[L'_j]}$ for each $j \in [m]$ is equivalent to (3.11).

For the experiments, we develop MILP formulations (A.2) and (A.3) applying the ideas of (IPPR1) and (IPPR2) to (3.11), which is the problem where multiple λ variables are used for the same vertex $\bar{\boldsymbol{z}}$. Formulations (A.2) and (A.3) apply to the cases where HPs are regular and non-regular, respectively. The full descriptions of these formulations are available in Appendix A.

For our numerical experiment, we first train tree ensemble models with linear regression trees for the diabetes data set from the UCI machine learning repository [6] and for the California house price data set [24]. We use the training algorithm available at https://github.com/cerlymarco/linear-tree. We choose different maximum depths ($D = 2, 3, 4$) and vary the number of trees in the ensemble ($T = 5, 10, 15, 20$). Finally, we solve the instances of TEO with formulation (A.3) applied to their natural non-regular HPs and with formulation (A.2) applied to the refined RHPs described above.

Table 3 summarizes our experimental results for both data sets. Each row in the table indicates what data set is being considered, together with the number n of its input variables, the maximum depth D of trees, and the number of trees n_w in the ensemble. It then displays the numbers of hyper-rectangles used in each of the formulations and the time it takes to solve them. We observe that the refined HPs require the introduction of a significant number of hyper-rectangles; this number increases as n or D increases. This is in agreement with our discussion in Example 2, where we showed that the number of hyper-rectangles for an RHP may be exponentially larger than those in a non-regular HP. In particular, the reason that the number of hyper-rectangles required in RHPs increases quickly when n or D becomes large is precisely the reason that it increases quickly when the parameters n or K of Example 2 become large. In Remark 3.2, we gave a decision tree for Example 2 where D is a linear function of n and K. Therefore, it follows easily that, for this example, the number of hyper-rectangles in the non-regular HP is bounded from above by $nD + 1$ while the number of hyper-rectangles in an RHP is $(D - n + 2)^n$, which is an exponential blowup. Table 3 shows that this increase is not only an artifact of the special example setting but is also observed for tree ensembles which fit real data sets. Table 3 also establish that there is a computational penalty for the increase in number of partition elements as solution times increase significantly for RHP as n or D become larger.

4. Conclusion. In this paper, we construct piecewise polyhedral relaxations (PPRs) of multilinear optimization problems over (axis-parallel) hyper-rectangular partitions (HPs). We provide a new formulation for PPRs over regular HPs (RHPs) using linking constraints. These constraints improve the formulations based on individual polyhedral relaxations found in the literature. We implement our relaxation inside the open-source MINLP solver ALPINE and show that the proposed change significantly improves ALPINE's performance on a variety of multilinear and polynomial optimization problem instances from Los Alamos MINLPLib. In short, we show that the new formulation can solve the same number of instances in an order-of-magnitude less time and can solve more than twice as many instances if given the same amount of time. We also provide the first MILP formulation for PPRs over non-regular HPs. Finally, we perform computational experiments that show that non-regular HPs formulations capture decision tree structure in a more compact formulation. As a result, they are typically an order-of-magnitude faster to solve than the formulations based on RHPs.

Appendix A. Formulations Used for Experiments in section 3.5.

In this section, we provide a formulation for the following problem:

$$\max \quad \boldsymbol{c}_{\boldsymbol{z}}^{\intercal}\boldsymbol{z} + \boldsymbol{c}_{\boldsymbol{w}}^{\intercal}\boldsymbol{w} \tag{A.1a}$$

$$\text{s.t.} \quad A_{\boldsymbol{z}}\boldsymbol{z} + A_{\boldsymbol{w}}\boldsymbol{w} \leq \boldsymbol{b}, \tag{A.1b}$$

$$\begin{pmatrix} \boldsymbol{z}_{I_j} \\ w_j \end{pmatrix} \in \bigcup_{i\in[L_j]} \bar{Q}_{j,i}, \qquad \forall j \in [n_w]. \tag{A.1c}$$

We first provide an MILP formulation (A.2) for (A.1) when HPs are regular but their discretization points for each axis are not same. For all $j \in [n_w]$, we define $\bar{\mathcal{Z}}_j := \{(\bar{\boldsymbol{z}}, \bar{w})\}_{i\in[L_j],(\bar{\boldsymbol{z}},\bar{w})\in \operatorname{vert} \bar{Q}_{j,i}}$ which is the collection of extreme points in the j^{th} HP.

$$\max \quad \boldsymbol{c}_{\boldsymbol{z}}^{\intercal}\boldsymbol{z} + \boldsymbol{c}_{\boldsymbol{w}}^{\intercal}\boldsymbol{w} \tag{A.2a}$$

$$\text{s.t.} \quad A_{\boldsymbol{z}}\boldsymbol{z} + A_{\boldsymbol{w}}\boldsymbol{w} \leq \boldsymbol{b}, \tag{A.2b}$$

$$\boldsymbol{z}_{I_j} = \sum_{(\bar{\boldsymbol{z}},\bar{w})\in\bar{\mathcal{Z}}_j} \bar{\boldsymbol{z}}\lambda^j_{\bar{\boldsymbol{z}},\bar{w}}, \qquad \forall j\in[n_w], \tag{A.2c}$$

$$w_j = \sum_{(\bar{\boldsymbol{z}},\bar{w})\in\bar{\mathcal{Z}}_j} \bar{w}\lambda^j_{\bar{\boldsymbol{z}},\bar{w}}, \qquad \forall j\in[n_w], \tag{A.2d}$$

$$\sum_{(\bar{\boldsymbol{z}},\bar{w})\in\bar{\mathcal{Z}}_j:\bar{\boldsymbol{z}}_{(v)}\le d_{v,k_2}} \lambda^j_{\bar{\boldsymbol{z}},\bar{w}} \le \sum_{k=1}^{k_2} x_{v,k}, \qquad \forall j\in[n_w],\forall v\in I_j,\forall k_2\in[D_v-2], \tag{A.2e}$$

$$\sum_{(\bar{\boldsymbol{z}},\bar{w})\in\bar{\mathcal{Z}}_j:\bar{\boldsymbol{z}}_{(v)}\ge d_{v,k_1+1}} \lambda^j_{\bar{\boldsymbol{z}},\bar{w}} \le \sum_{k=k_1}^{D_v-1} x_{v,k}, \qquad \forall j\in[n_w],\forall v\in I_j, \; \forall k_1\in[D_v-1]\setminus\{1\}, \tag{A.2f}$$

$$\sum_{(\bar{\boldsymbol{z}},\bar{w})\in\bar{\mathcal{Z}}_j:\bar{\boldsymbol{z}}_{(v)}\in[d_{v,k_1+1},d_{v,k_2}]} \lambda^j_{\bar{\boldsymbol{z}},\bar{w}} \le \sum_{k=k_1}^{k_2} x_{v,k}, \qquad \forall j\in[n_w],\forall v\in I_j, \; \forall k_1<k_2\in[D_v-2]\setminus\{1\}, \tag{A.2g}$$

$$\boldsymbol{\lambda^j} = \{\lambda^j_{\bar{\boldsymbol{z}},\bar{w}}\}_{(\bar{\boldsymbol{z}},\bar{w})\in\bar{\mathcal{Z}}_j} \in \Delta^{|\bar{\mathcal{Z}}_j|}, \qquad \forall j\in[n_w], \tag{A.2h}$$

$$\boldsymbol{x_v} \in \Delta^{D_v-1}_{0,1}, \qquad \forall v\in[n]. \tag{A.2i}$$

We next provide an MILP formulation for (A.1) when HPs are non-regular. To relate $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ when HPs are non-regular, we construct bipartite graph $G_j = (U_j, V_j, E_j)$ where $U_j = \bar{\mathcal{Z}}_j$, $V_j = [L_j]$, and $E_j = \{(\bar{\boldsymbol{z}}, \bar{w}, i) \in U_j \times V_j \mid (\bar{\boldsymbol{z}}, \bar{w}) \in \operatorname{vert} \bar{Q}_{j,i}\}$ for all $j \in [n_w]$.

$$\max \quad \boldsymbol{c_z}^\intercal \boldsymbol{z} + \boldsymbol{c_w}^\intercal \boldsymbol{w} \tag{A.3a}$$

$$\text{s.t.} \quad A_{\boldsymbol{z}}\boldsymbol{z} + A_{\boldsymbol{w}}\boldsymbol{w} \le \boldsymbol{b}, \tag{A.3b}$$

$$(\boldsymbol{z}_{I_j}, w_j) = \sum_{(\bar{\boldsymbol{z}},\bar{\boldsymbol{w}})\in\bar{\boldsymbol{\mathcal{Z}}}_j} (\bar{\boldsymbol{z}},\bar{w})\lambda^j_{\bar{z},\bar{w}}, \qquad \forall j\in[n_w], \tag{A.3c}$$

$$\sum_{e=(u',v')\in E_j:u'=u} h_{j,e} = \lambda^j_u, \qquad \forall j\in[n_w],\forall u\in U_t, \tag{A.3d}$$

$$\sum_{e=(u',v')\in E_j:v'=v} h_{j,e} = y_{j,i}, \qquad \forall j\in[n_w],\forall v\in V_j, \tag{A.3e}$$

$$\sum_{i\in[L_j]:k_1\le k_1(j,i,v),k_2(j,i,v)\le k_2} y_{j,i} \le \sum_{k=k_1}^{k_2} x_{v,k}, \qquad \forall j\in[n_w],\forall v\in[n], \; \forall k_1\le k_2\in[D_v-1], \tag{A.3f}$$

$$\boldsymbol{\lambda^j} = \{\lambda^j_{\bar{\boldsymbol{z}}}\}_{\bar{\boldsymbol{z}}\in\bar{\mathcal{Z}}_j} \in \Delta^{|\bar{\mathcal{Z}}_j|}, \qquad \forall j\in[n_w], \tag{A.3g}$$

$$h_{j,e} \ge 0, \qquad \forall j\in[n_w],\forall e\in E_j, \tag{A.3h}$$

$$\text{(A.3i)} \qquad \boldsymbol{y_j} \in \Delta^{L_j}, \qquad \forall j \in [n_w],$$

$$\text{(A.3j)} \qquad \boldsymbol{x_v} \in \Delta_{0,1}^{D_v - 1}, \qquad \forall v \in [n].$$

Acknowledgments. The research presented in this paper was supported by grants 1727989 and 1917323 from the National Science Foundation Division of Civil, Mechanical and Manufacturing Innovation.

REFERENCES

[1] X. Bao, A. Khajavirad, N. V. Sahinidis, and M. Tawarmalani, *Global optimization of nonconvex problems with multilinear intermediates*, Mathematical Programming Computation, 7 (2015), pp. 1–37, https://doi.org/10.1007/s12532-014-0073-z.

[2] Y.-C. Chen and V. V. Mišić, *Decision forest: A nonparametric approach to modeling irrational choice*, Management Science, (2022), https://doi.org/10.1287/mnsc.2021.4256.

[3] Y. Crama, *Concave extensions for nonlinear 0–1 maximization problems*, Mathematical Programming, 61 (1993), pp. 53–60, https://doi.org/10.1007/BF01582138.

[4] A. Dobra and J. Gehrke, *Secret: A scalable linear regression tree algorithm*, in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 2002, pp. 481–487, https://doi.org/abs/10.1145/775047.775117.

[5] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213, https://doi.org/10.1007/s101070100263.

[6] D. Dua and C. Graff, *UCI machine learning repository*, 2017, https://archive.ics.uci.edu/ml.

[7] I. Dunning, J. Huchette, and M. Lubin, *JuMP: A modeling language for mathematical optimization*, SIAM Review, 59 (2017), pp. 295–320, https://doi.org/10.1137/15M1020575.

[8] C. D'Ambrosio, A. Lodi, and S. Martello, *Piecewise linear approximation of functions of two variables in MILP models*, Operations Research Letters, 38 (2010), pp. 39–46, https://doi.org/10.1016/j.orl.2009.09.005.

[9] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. De Wolf, *Exponential lower bounds for polytopes in combinatorial optimization*, Journal of the ACM (JACM), 62 (2015), pp. 1–23, https://doi.org/10.1145/2716307.

[10] A. Gleixner, M. Bastubbe, L. Eifler, T. Gally, G. Gamrath, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. Lübbecke, S. J. Maher, M. Miltenberger, B. Müller, M. Pfetsch, C. Puchert, D. Rehfeldt, F. Schlösser, C. Schubert, F. Serrano, Y. Shinano, J. M. Viernickel, M. Walter, F. Wegscheider, J. T. Witt, and J. Witzig, *The SCIP Optimization Suite 6.0*, Tech. Report 18-26, Zuse Institute Berlin, 2018, https://nbn-resolving.de/urn:nbn:de:0297-zib-69361.

[11] Gurobi Optimization, LLC, *Gurobi optimizer reference manual*, 2021, https://www.gurobi.com.

[12] T. He and M. Tawarmalani, *Extracting structure from functional expressions for continuous and discrete relaxations of MINLP*. arXiv, 2022, https://arxiv.org/abs/2205.01442.

[13] T. He and M. Tawarmalani, *MIP relaxations of composite functions*. Working paper, 2022.

[14] A. J. Hoffman, *Total unimodularity and combinatorial theorems*, Linear Algebra and its Applications, 13 (1976), pp. 103–108, https://doi.org/10.1016/0024-3795(76)90047-1.

[15] J. Huchette and J. P. Vielma, *Nonconvex piecewise linear functions: Advanced formulations and simple modeling tools*, Operations Research, (2022), https://doi.org/10.1287/opre.2019.1973.

[16] J. Kim, J.-P. P. Richard, and M. Tawarmalani, *A reciprocity between tree ensemble optimization and multilinear optimization*. Available at Optimization Online, 2022, http://www.optimization-online.org/DB_HTML/2022/03/8828.html.

[17] T. Kis and M. Horváth, *Ideal, non-extended formulations for disjunctive constraints admitting a network representation*, Mathematical Programming, (2021), https://doi.org/10.1007/s10107-021-01652-z.

[18] J. Luedtke, M. Namazifar, and J. Linderoth, *Some results on the strength of relaxations of multilinear functions*, Mathematical Programming, 136 (2012), pp. 325–351, https://doi.org/10.1007/s10107-012-0606-z.

[19] G. P. McCormick, *Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems*, Mathematical programming, 10 (1976), pp. 147–175, https://doi.org/10.1007/BF01580665.

[20] R. Misener and C. A. Floudas, *ANTIGONE: algorithms for continuous/integer global opti-*

mization of nonlinear equations, Journal of Global Optimization, 59 (2014), pp. 503–526.

[21] V. V. Mišić, *Optimization of tree ensembles*, Operations Research, 68 (2020), pp. 1605–1624, https://doi.org/10.1287/opre.2019.1928.

[22] H. Nagarajan, M. Lu, S. Wang, R. Bent, and K. Sundar, *An adaptive, multivariate partitioning algorithm for global optimization of nonconvex programs*, Journal of Global Optimization, 74 (2019), pp. 639–675, https://doi.org/10.1007/s10898-018-00734-1.

[23] H. Nagarajan, M. Lu, E. Yamangil, and R. Bent, *Tightening McCormick relaxations for nonlinear programs via dynamic multivariate partitioning*, in International conference on principles and practice of constraint programming, Springer, 2016, pp. 369–387, https://doi.org/10.1007/978-3-319-44953-1_24.

[24] R. K. Pace and R. Barry, *Sparse spatial autoregressions*, Statistics & Probability Letters, 33 (1997), pp. 291–297, https://doi.org/10.1016/S0167-7152(96)00140-X.

[25] M. Padberg, *The boolean quadric polytope: some characteristics, facets and relatives*, Mathematical programming, 45 (1989), pp. 139–172, https://doi.org/10.1007/BF01589101.

[26] J. R. Quinlan, *Learning with continuous classes*, in 5th Australian joint conference on artificial intelligence, vol. 92, World Scientific, 1992, pp. 343–348, https://doi.org/10.1142/9789814536271.

[27] A. Schrijver, *Short proofs on the matching polyhedron*, Journal of Combinatorial Theory, Series B, 34 (1983), pp. 104–108, https://doi.org/10.1016/0095-8956(83)90011-4.

[28] K. Sundar, H. Nagarajan, J. Linderoth, S. Wang, and R. Bent, *Piecewise polyhedral formulations for a multilinear term*, Operations Research Letters, 49 (2021), pp. 144–149, https://doi.org/10.1016/j.orl.2020.12.002.

[29] K. Sundar, S. Sanjeevi, and H. Nagarajan, *Sequence of polyhedral relaxations for nonlinear univariate functions*, Optimization and Engineering, (2021), pp. 1–18, https://doi.org/10.1007/s11081-021-09609-z.

[30] M. Tawarmalani and N. V. Sahinidis, *Convex extensions and envelopes of lower semi-continuous functions*, Mathematical Programming, 93 (2002), pp. 247–263, https://doi.org/10.1007/s10107-002-0308-z.

[31] J. P. Vielma, S. Ahmed, and G. Nemhauser, *Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions*, Operations Research, 58 (2010), pp. 303–315, https://doi.org/10.1287/opre.1090.0721.

[32] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming, 106 (2006), pp. 25–57, https://doi.org/10.1007/s10107-004-0559-y.