MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Recursive McCormick Linearization of Multilinear Programs

Raghunathan, Arvind; Cardonha, Carlos; Bergman, David; Nohra, Carlos J.

TR2025-098 July 01, 2025

Abstract

Linear programming (LP) relaxations are widely employed in exact solution methods for multilinear programs (MLPs). These relaxations can be obtained by using Recursive McCormick Linearizations (RMLs), where an MLP is linearized by iteratively substituting bilinear products with artificial variables and constraints. This article introduces a systematic approach to identifying RMLs. We focus on identifying RMLs with a small number of artificial variables and strong LP bounds. We present a novel mechanism for representing all the possible RMLs, which we use to design an exact mixed-integer programming (MIP) formulation to identify minimum-size RMLs; this problem is NP-hard in general, but we show that it is fixed-parameter tractable if each monomial is composed of at most three variables. Moreover, we explore the structural properties of our formulation to derive an exact MIP model that identifies RMLs of a given size with the best-possible LP relaxation bound. We test our algorithms by conducting numerical experiments on a large collection of MLPs. Numerical results indicate that the RMLs obtained with our algorithms can be significantly smaller than those derived from heuristic or greedy approaches, leading, in many cases, to tighter LP relaxation bounds. Moreover, our linearization strategies can be used to reformulate MLPs as QCPs, which can then be efficiently solved using state-of-the-art solvers for QCPs. This QCP-based solution approach is highly beneficial for hard MLP instances.

INFORMS J Computing 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Recursive McCormick Linearization of Multilinear Programs

Carlos Cardonha
University of Connecticut, Storrs, CT 06269, carlos.cardonha@uconn.edu

Arvind U Raghunathan
Mitsibushi Electric Research Laboratories, Cambridge, MA 02139, raghunathan@merl.com

David Bergman
University of Connecticut, Storrs, CT 06269, david.bergman@uconn.edu

Carlos J. Nohra
Amadeus North America, Irving, TX 75062.

Abstract. Linear programming (LP) relaxations are widely employed in exact solution methods for multilinear programs (MLPs). These relaxations can be obtained by using Recursive McCormick Linearizations (RMLs), where an MLP is linearized by iteratively substituting bilinear products with artificial variables and constraints. This article introduces a systematic approach to identifying RMLs. We focus on identifying RMLs with a small number of artificial variables and strong LP bounds. We present a novel mechanism for representing all the possible RMLs, which we use to design an exact mixed-integer programming (MIP) formulation to identify minimum-size RMLs; this problem is NP-hard in general, but we show that it is fixed-parameter tractable if each monomial is composed of at most three variables. Moreover, we explore the structural properties of our formulation to derive an exact MIP model that identifies RMLs of a given size with the best-possible LP relaxation bound. We test our algorithms by conducting numerical experiments on a large collection of MLPs. Numerical results indicate that the RMLs obtained with our algorithms can be significantly smaller than those derived from heuristic or greedy approaches, leading, in many cases, to tighter LP relaxation bounds. Moreover, our linearization strategies can be used to reformulate MLPs as QCPs, which can then be efficiently solved using state-of-the-art solvers for QCPs. This QCP-based solution approach is highly beneficial for hard MLP instances.

Key words: Multilinear Programs, McCormick Linearization, Global Optimization, Parameterized Complexity

1. Introduction

This article introduces new techniques for linearizing unconstrained *multilinear programs* (MLPs) defined over $\Omega = [0,1]^n$ or $\Omega = \{0,1\}^n$. An MLP is formulated as

$$\min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i \prod_{j\in J_i} x_j. \tag{1}$$

We use $\boldsymbol{x} = (x_1,\ldots,x_n)$ to denote a vector in Ω. Function $f(\boldsymbol{x})$ consists of m monomials. Each monomial $\alpha_i f_i(\boldsymbol{x})$, $i \in [m]$, is composed of a coefficient $\alpha_i \in \mathbb{R}$ and a term $f_i(\boldsymbol{x}) \coloneqq \prod_{j\in J_i} x_j$, i.e., $f_i(\boldsymbol{x})$ is the product of the variables whose indices belong to a subset J_i of $[n]$. We assume w.l.o.g. that the term $f_i(\boldsymbol{x})$ is unique to the monomial $\alpha_i f_i(\boldsymbol{x})$ (as monomials sharing the same term can be aggregated).

EXAMPLE 1. Consider the MLP $\min_{\boldsymbol{x}\in[0,1]^4} f(\boldsymbol{x}) = x_1x_2x_3 - x_2x_3x_4 - x_1x_3x_4$ consisting of $m = 3$ monomials defined over $n = 4$ variables with domain $\Omega = [0,1]$. Monomial $\alpha_1 f_1(\boldsymbol{x}) = x_1x_2x_3$ is described by the coefficient $\alpha_1 = 1$ and the set of variables $J_1 = \{1,2,3\}$, which is associated with the term $f_1(\boldsymbol{x}) = x_1x_2x_3$.

MLPs arise in a wide variety of applications such as circuit layout design (Boros et al. (1999)), facility location (Jakob and Pruzan (1983)) and statistical mechanics (Bernasconi (1987)).

Branch-and-bound methods can directly be used to solve (1) to global optimality. The efficiency of these methods is greatly influenced by the computational cost and tightness of the relaxations used to bound the problem. MLPs can be bounded using Linear Programming (LP), Mixed-Integer Linear Programming (MILP) and Semidefinite Programming (SDP) relaxations. LP relaxations are typically constructed in two steps. In the first step, factorable programming techniques are used to derive an initial LP relaxation of the MLP in a higher-dimensional space (McCormick (1976), Sherali and Wang (2001), Tawarmalani and Sahinidis (2004)). In the second step, this initial LP relaxation is tightened by adding different classes of valid inequalities, such as facets of the convex and concave envelopes of the multilinear functions contained in the MLP (Bao et al. (2009, 2015), Misener et al. (2015)), Reformulation-Linearization Technique (RLT) inequalities (Zorn and Sahinidis (2014), Dalkiran and Sherali (2016)), 2-links inequalities (Crama and Rodríguez-Heck (2017)), running intersection inequalities (Del Pia et al. (2020)) and extended flower inequalities (Khajavirad (2023)). MILP relaxations can be constructed by using piece-wise linear formulations of multilinear terms (Sundar et al. (2021), Kim et al. (2022)). When the original MLP contains discrete variables, an MILP relaxation can alternatively be obtained by re-imposing integrality requirements on an existing LP relaxation of the MLP (Kılınç and Sahinidis (2018)). SDP relaxations can be derived using the theory of moments and sum-of-squares approaches (Lasserre (2001)).

An alternative to directly solving (1) is to reformulate it into a different class of problem, which may be easier to solve. To this end, additional variables and constraints are introduced to obtain an equivalent problem with a linear or quadratic objective function. Under these approaches, some or all of the non-linearity is moved from the objective to the newly introduced constraints. In the literature, these reformulations are referred to as linearization or quadratization approaches (Anthony et al. (2017), Rodríguez-Heck (2018), Dalkiran and Ghalami (2018), Elloumi et al. (2021), Karia et al. (2022)). When all variables in (1) are binary, the problem can be reduced to a binary linear program via linearization (Glover and Woolsey (1974)).

A widely used linearization strategy for (1) involves replacing bilinear products x_ix_j, appearing in one or more multilinear terms, with artificial variables $y_{\{i,j\}}$. By iteratively applying such operations, the MLP can be reformulated into an equivalent problem with a linear objective function and nonconvex quadratic constraints of the form $y_{\{i,j\}} = x_ix_j$. By replacing these nonlinear constraints with McCormick inequalities (McCormick (1976)), an LP relaxation of (1) is obtained. We refer to linearization strategies following the aforementioned procedure as *Recursive McCormick Linearizations* (RMLs). The number of artificial variables and the quality of the LP relaxation bound varies across different RMLs, as illustrated in Example 2.

EXAMPLE 2 (QUALITY OF AN RML). Figures 1 and 2 depict two RMLs for the MLP shown in Example 1. RML 1 uses ten variables in total (with six artificial variables) and delivers an LP bound of $\frac{-4}{3}$. In contrast, RML 2 uses five artificial variables and has an LP bound of -1.

$$\underbrace{\underbrace{x_1x_2}_{}x_3}_{y_{\{1,2,3\}}} \quad \underbrace{\underbrace{x_2x_3}_{}x_4}_{y_{\{2,3,4\}}} \quad \underbrace{\underbrace{x_1x_3}_{}x_4}_{y_{\{1,3,4\}}}$$

$$y_{\{1,2\}}x_3 \quad y_{\{2,3\}}x_4 \quad y_{\{1,3\}}x_4$$

Figure 1 RML with 10 variables and LP bound $\frac{-4}{3}$.

$$\underbrace{x_2\underbrace{x_1x_3}_{}}_{y_{\{1,2,3\}}} \quad \underbrace{x_2\underbrace{x_3x_4}_{}}_{y_{\{2,3,4\}}} \quad \underbrace{x_1\underbrace{x_3x_4}_{}}_{y_{\{1,3,4\}}}$$

$$x_2y_{\{1,3\}} \quad x_2y_{\{3,4\}} \quad x_1y_{\{3,4\}}$$

Figure 2 RML with 9 variables and LP bound -1.

In this article, we present a systematic study of RMLs focused on the size and tightness of the resulting LP relaxation. In particular, we introduce exact approaches for identifying RMLs that have (i) the minimum number of artificial variables; and (ii) the tightest LP relaxation bound given a constraint on the number of artificial variables. Our theoretical and algorithmic contributions are:

- **Minimum-Size RML:** We show that the identification of minimum-size RMLs is NP-hard, and we present a fixed-parameter tractable algorithm for the special case of the problem where all monomials have degree at most three. Furthermore, we investigate scenarios where a greedy approach to the problem delivers arbitrarily poor results. Finally, we propose an exact MIP model for finding minimum-size RMLs.
- **Best-Bound RML**: We introduce an exact MIP model for finding best-bound RMLs of any given size. Our results rely on transforming a two-level MIP formulation into a single-level MIP based on the bounds we derive for dual variables of the inner-level sub-problem.

Our algorithms can be used in two different ways. First, our RMLs can be used within a global optimization solver to obtain initial LP relaxations with a reduced size and tighter bounds. As indicated previously, these initial LP relaxations can be further tightened by adding valid inequalities. Second, our RMLs can be used to reformulate (1) into a quadratically-constrained problem (QCP), which can then be solved using a global optimization solver for QCPs.

To demonstrate the benefits of our approach, we conduct numerical experiments on a large set of MLP instances. Our results suggest that our minimum-size RMLs can be significantly smaller than the RMLs obtained using heuristic or greedy approaches. Similarly, our best-bound RMLs can significantly improve the LP relaxation bounds. Moreover, we use our RMLs to reformulate all MLP instances in the test set as QCPs, which we then solve with the global solver GUROBI. We compare this solution approach with another one based on directly solving the MLPs with the global solver BARON. As our results indicate, for hard instances, using our linearization strategies to reformulate the MLPs as QCPs, and solving the resulting QCPs with GUROBI can be significnatly faster than solving the original MLPs with BARON.

The closest reference to our work is the recent paper by Elloumi and Verchère (2023), which proposes algorithms for identifying minimum-size RMLs and best-bound RMLs using a different modeling approach. Namely, whereas the model in Elloumi and Verchère (2023) has variables representing all possible linearizations, our model decomposes by bilinear terms, so our formulation is more compact and scalable for monomials of large degree. Moreover, the differences in the modeling approaches are relevant for the design of an exact MIP model for finding best-bound RMLs. Namely, both papers rely on single-level reformulations of bi-level optimization problems, but whereas the reformulation of Elloumi and Verchère (2023) is straightforward, ours requires more careful and sophisticated analysis. Our numerical experiments show that our algorithms outperform the algorithms by Elloumi and Verchère (2023), mainly because our models obtain better dual bounds for harder instances. Verma and Lewis (2020) also study minimum-size RMLs, but their algorithm is tailored for monomials of degree four, whereas ours can be applied to arbitrary MLPs. Other references related to our work include the articles by Anthony et al. (2016) and Boros et al. (2020), which focus on identifying upper and lower bounds for the size of quadratizations.

Our results extend to the cases where all variables are subject to general box constraints, as multilinear functions are closed under affine variable transformations. Additionally, our algorithms can also be used to solve general polynomial optimization problems over a box, including cases with polynomial constraints.

The remainder of this article is organized as follows. Section 2 formalizes the problem and introduces the notation. Sections 3 and 4 present our results involving minimum-size RMLs and best-bound RMLs, respectively. Section 5 presents our numerical studies. Finally, Section 6 concludes the article.

2. Linearization of Multilinear Programs

The standard algorithm for linearizing an MLP is an iterative procedure that reduces the degree of one or more multilinear terms in each step by replacing bilinear terms with artificial variables. Consider any index $i \in [m]$ and the corresponding term $f_i(\boldsymbol{x}) = \prod_{j \in J_i} x_j$. For any pair of indices $j_1, j_2 \in J_i$, we can replace the bilinear product $x_{j_1}x_{j_2}$ with the variable $y_{\{j_1,j_2\}}$ and rewrite $f_i(\boldsymbol{x})$ as $f_i(\boldsymbol{x}) = y_{\{j_1,j_2\}} \prod_{j \in J_i \setminus \{j_1,j_2\}} x_j$. This reformulation does not eliminate nonlinearity, but we can use McCormick convex and concave envelopes (see McCormick (1976)) to obtain a polyhedral relaxation of this expression:

$$
\begin{aligned}
y_{\{j_1,j_2\}} & & \geq 0 \\
y_{\{j_1,j_2\}} & - x_{j_1} - x_{j_2} + 1 & \geq 0 \\
y_{\{j_1,j_2\}} & - x_{j_1} & \leq 0 \\
y_{\{j_1,j_2\}} & - x_{j_2} & \leq 0
\end{aligned}
$$

We denote the McCormick inequality system that linearizes the bilinear product $x_{j_1}x_{j_2}$ by introducing an artificial variable $y_{\{j_1,j_2\}}$ and the convex and concave envelopes in (2) as $\mathcal{E}(\mathsf{t})$ with $\mathsf{t} = (\{j_1\}, \{j_2\}, \{j_1, j_2\})$. This procedure can be recursively applied to the remaining bilinear products of original and artificial variables until $f_i(\boldsymbol{x})$ is completely linearized. In the end, we obtain a set of artificial variables y_J representing the product of all variables in $J \subseteq [n]$ for a collection of sets that includes J_i for each $i \in [m]$. For ease of notation, we also refer to the variable x_j as $y_{\{j\}}$.

2.1. Recursive McCormick Relaxation (RML)

For any $i \in [m]$, let $\mathcal{N}_i := \{J : J \subseteq J_i, J \neq \emptyset\}$ be the family of non-empty subsets of indices of the variables in monomial i, and let $\mathcal{N} = \bigcup_{i \in [m]} \mathcal{N}_i$. For any J'' in $\mathcal{N}$ such that $|J''| \geq 2$, a *triple* $\mathsf{t} = (J, J', J'')$ describes a partition of J'' into two non-empty sets J and J'. We assume that the first two elements of any triple are arranged in lexicographical order. In this way, we can uniquely define $\mathsf{tail1}(\mathsf{t})$, $\mathsf{tail2}(\mathsf{t})$, and $\mathsf{head}(\mathsf{t})$ as the first, second, and third elements of t, respectively. Finally, let $\mathcal{T}_i := \{\mathsf{t} : \mathsf{head}(\mathsf{t}) \in \mathcal{N}_i\}$ and $\mathcal{T} = \bigcup_{i \in [m]} \mathcal{T}_i$ be the set of all possible triples associated with $\mathcal{N}_i$ and $\mathcal{N}$, respectively, and let $\mathsf{tails}(\mathsf{t}) := \{\mathsf{tail1}(\mathsf{t}), \mathsf{tail2}(\mathsf{t})\}$.

DEFINITION 1. A *Proper Triple Set* for an MLP is a set of triples $T \subseteq \mathcal{T}$ for which there exists a subset $T' \subseteq T$ satisfying the following conditions:

RMP 1 For every monomial $\alpha_i f_i(\boldsymbol{x})$, if the set J_i with the indices of the variables composing $f_i(\boldsymbol{x})$ is such that $|J_i| > 1$, then J_i is equal to $\mathsf{head}(\mathsf{t})$ for at least one triple $\mathsf{t} \in T'$; and

RMP 2 If a set $J \subseteq \mathcal{N}$ such that $|J| > 1$ is equal to $\mathsf{tail1}(\mathsf{t})$ or $\mathsf{tail2}(\mathsf{t})$ for some triple $\mathsf{t} \in T'$, then J is equal to $\mathsf{head}(\mathsf{t}')$ for some $\mathsf{t}' \neq \mathsf{t}$ in T'.

Condition **RMP 1** enforces the linearization of all monomials of two or more variables. Condition **RMP 2** extends **RMP 1** to artificial variables, which always represent the product of two or more original variables. Our definition of proper triple sets enforces the existence of at least one full linearization and allows the incorporation of artificial variables with no associated linearization; these variables are not necessary, but they may help enhance relaxation bounds. A proper triple set T defines an RML of an MLP over the set of variables y_J for each J in $\{\mathsf{head}(\mathsf{t}) : \mathsf{t} \in T\}$ and subject to the constraints of $\mathcal{E}(\mathsf{t})$ for each t in T:

$$\begin{aligned} \min \quad & \sum_{i\in[m]} \alpha_i y_{J_i} & \\ \text{s.t.} \quad & \mathcal{E}(\mathsf{t}), & \forall \mathsf{t} \in T \\ & y_J \in [0,1], & \forall J \in \mathcal{N}. \end{aligned} \tag{2}$$

The cardinality of T defines the size of the respective RML (2). In particular, minimum-size RMLs are minimal proper triple sets. We consider non-minimal proper triple sets when identifying best-bound LP relaxation linearizations (see Section 4). Lastly, an RML can also be used to derive a quadratization of an MLP. We obtain the following quadratically-constraint problem (QCP) of an MLP from an RML T:

$$\begin{aligned} \min \quad & \sum_{i\in[m]} \alpha_i y_{J_i} & \\ \text{s.t.} \quad & y_{J_1\cup J_2} = y_{J_1} y_{J_2}, & \forall (J_1, J_2, J_1 \cup J_2) \in T \\ & y_J \in \Omega, & \forall J \in \mathcal{N}. \end{aligned} \tag{3}$$

The domain Ω of variable y_J is set to $[0,1]$ or $\{0,1\}$ based on the domains of the variables composing J.

2.2. Sequential RML

Algorithm 1 describes `Seq`, the *recursive arithmetic interval* RML strategy by Ryoo and Sahinidis (2001). `Seq` is an iterative algorithm that, in each step, identifies a pair of (original or artificial)

Algorithm 1: Sequential RML (`Seq`)

$T := \emptyset$ *Set of triples*
for $i \in [m]$ **do**
  $A_i := \{\{j\} : j \in J_i\}$ *Family of singleton sets containing the indices of the variables composing $f_x(\boldsymbol{x})$*
while $\exists i \in [m] : |A_i| > 1$ **do**
  Pick $J, J' \in A_i$ **for some** A_i **such that** $|A_i| > 1$ *Select an arbitrary bilinear term from an arbitrary monomial*
  $J'' := J \cup J'$
  $T := T \cup \{(J, J', J'')\}$
  for $i' \in [m]$ **do**
    if $\{J, J'\} \subseteq A_{i'}$ **then**
      $A_{i'} := A_{i'} \setminus \{J, J'\}$ *Remove sets J and J' from $A_{i'}$*
      $A_{i'} := A_{i'} \cup \{J''\}$ *Add set $J \cup J'$ to $A_{i'}$*

variables y_J and $y_{J'}$ occurring in the same term, where $J \cap J' = \emptyset$, and replaces the bilinear product $y_J y_{J'}$ for a new artificial variable $y_{J\cup J'}$. This substitution is applied to *all* multilinear terms containing $y_J y_{J'}$. Example 3 shows that the linearization generated by `Seq` depends on the variable ordering it uses to select bilinear terms.

EXAMPLE 3. `Seq` yields the linearization depicted in Figure 1 for $f(\boldsymbol{x}) = x_1x_2x_3 - x_2x_3x_4 - x_1x_3x_4$ if it adopts the ordering (x_1, x_2, x_3, x_4), which leads to the substitution of the bilinear terms x_1x_2, x_2x_3, and x_1x_3, in this order. In contrast, `Seq` obtains the linearization in Figure 2 if it uses the ordering (x_3, x_4, x_1, x_2) instead; first, x_3x_4 is replaced in the last two monomials, and then x_1x_3 is replaced in the first.

2.3. Related Concepts in the Literature

RMLs are related to linearization and quadratization techniques explored in the literature. For example, a triple can be interpreted as a reduction operation in the method presented by Buchheim and Rinaldi (2008). More recently, Anthony et al. (2017) presented the concept of pairwise covers for the quadratization of binary optimization problems. Namely, for any binary optimization problem, a pairwise cover consists of a collection H of subsets of indices in $[n]$ such that, for each monomial indexed by I, there is a pair of elements in H whose union equals I. This idea is related to RMLs, although a direct application of pairwise covers does not necessarily provide a full linearization for large-degree monomials; namely, some multilinear terms derived from this decomposition may not have a pairwise cover. Finally, and more closely related to our strategy, we have the quadratization scheme presented in Crama et al. (2022), which, similarly to us, differs from pairwise covers by enforcing the quadratization of all multilinear terms. The main difference between our strategy and the quadratization schemes is that each triple (J, J', J'') used in our linearizations describes a partition of J'' into non-empty sets J and J', whereas the quadratization scheme in Crama et al. (2022) (and Anthony et al. (2016)) allows $J \cap J' \neq \emptyset$. Therefore, our linearization can be interpreted as a restriction of the quadratization scheme in Crama et al. (2022).

3. Minimum-size Linearization

We investigate strategies for deriving minimum-size RMLs. We start with a simple yet suboptimal greedy approach and then proceed with an exact algorithm to find minimum-size RMLs. We also show that finding a minimum-size RML is NP-hard and that a special case of the problem is fixed-parameter tractable.

3.1. Greedy Linearization

Algorithm 2 describes `Greedy`, an RML strategy that selects a pair of variables y_J and $y_{J'}$ appearing together in as many monomials as possible in each iteration. Then, similarly to `Seq`, the bilinear product $y_J y_{J'}$ is replaced by $y_{J \cup J'}$ in each monomial where it occurs. `Greedy` is akin to the proce-

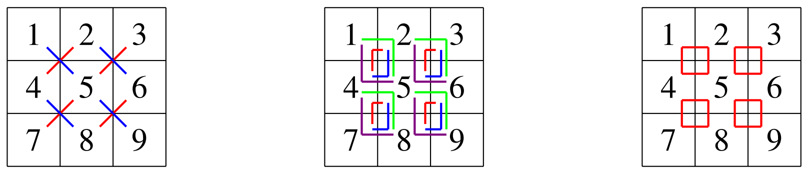
Algorithm 2: `Greedy`

$T := \emptyset$ *Set of triples*
for $i \in [m]$ **do**
    $A_i := \{\{j\} : j \in J_i\}$ *Family of singleton sets containing the indices of the variables composing* $f_x(\boldsymbol{x})$
while $\exists i \in [m] : |A_i| > 1$ **do**
    Pick J, J' **such that** $|\{i \in [m] : \{J, J'\} \in A_i\}|$ **is maximum**
    $J'' := J \cup J'$
    $T := T \cup \{(J, J', J'')\}$
    for $i' \in [m]$ **do**
        if $\{J, J'\} \subseteq A_{i'}$ **then**
            $A_{i'} := A_{i'} \setminus \{J, J'\}$ *Remove sets* J *and* J' *from* $A_{i'}$
            $A_{i'} := A_{i'} \cup \{J''\}$ *Add set* $J \cup J'$ *to* $A_{i'}$

dure described by Buchheim and Rinaldi (2008) for the quadratization of polynomial optimization problems. More recently, Rodríguez-Heck (2018) proposed heuristics for identifying small pairwise covers (Anthony et al. (2017)); in particular, the heuristics named “Most popular intersection first” and “Most popular pair first” explore the same ideas as the greedy algorithm, as they select sets for the pairwise cover based on how frequently they appear in the monomials.

This heuristic frequently performs well, but it can produce arbitrarily large RMLs (see Proposition 3). In particular, Example 4 shows why `Greedy` performs poorly in the `vision` instances (Crama and Rodríguez-Heck (2017)); this behavior is observed in our experiments.

EXAMPLE 4. The `vision` instances are multilinear polynomials with quadratic, cubic, and quartic terms. The variables represent cells in a grid. The quadratic, cubic, and quartic terms are associated with variables forming a diagonal, a right angle, and a square of adjacent cells, respectively. Figure 3 shows examples of terms in an instance of the problem defined over a 3-by-3 grid. An n-by-n instance has $2(n-1)^2$ quadratic terms, $4(n-1)^2$ cubic terms, and $(n-1)^2$ quartic terms. These instances admit a baseline RML with one artificial variable per term, starting with the



Quadratic terms   Cubic terms   Quartic terms

Figure 3 **All terms in a 3-by-3 example of the `vision` instances.**

quadratic ones and proceeding with the cubic and quartic terms. In contrast, `Greedy` adds an artificial variable $y_{\{i,i+1\}}$ for each $1 \leq i \leq n^2$ such that $i \mod n \neq 0$ first, representing pairs of cells that appear in the cubic and quartic terms but not in the quadratic terms. `Greedy` still needs to add one artificial variable for each term, so the first batch of artificial variables is added in addition to the same number of variables used by the baseline RML.

3.2. Exact Model

Formulation (4) is an MIP formulation for identifying minimum-size RMLs. The variables of (4) represent the selection of the triples composing a proper triple set T. Each $u_{i,\mathsf{t}}$ indicates whether triple t is used in the linearization of monomial $f_i(\boldsymbol{x})$, and v_{t} indicates whether t is used in the linearization of any monomial. The constraints (4b)-(4c) model conditions **RMP 1** and **RMP 2**, respectively. Namely, if the index set J_i of monomial i contains two or more elements, then $\mathsf{head}(\mathsf{t}) = J_i$ for at least one triple t in T. Similarly, if some index set J containing two or more elements is in $\mathsf{tails}(\mathsf{t})$ for some selected triple t, then there must exist another selected triple t' such that $J = \mathsf{head}(\mathsf{t}')$. The constraints (4d) couple variables $u_{i,\mathsf{t}}$ and v_{t}, i.e., if we use t to linearize one or more monomials, then we must set v_{t} to one. Finally, the objective function (4a) minimizes the number of triples used to linearize the entire MLP.

$$
\begin{aligned}
\min \; & \sum_{\mathsf{t} \in \mathcal{T}} v_{\mathsf{t}} && && \text{(4a)} \\
\text{s.t.} \; & \sum_{\mathsf{t} \in \mathcal{T}_i : \mathsf{head}(\mathsf{t}) = J_i} u_{i,\mathsf{t}} = 1 && \forall\, i \in [m] \text{ with } |J_i| > 1, && \text{(4b)} \\
& \sum_{\mathsf{t} \in \mathcal{T}_i : \mathsf{head}(\mathsf{t}) = J} u_{i,\mathsf{t}} = \sum_{\mathsf{t} \in \mathcal{T}_i : J \in \mathsf{tails}(\mathsf{t})} u_{i,\mathsf{t}} && \forall\, i \in [m], \forall\, J \in \mathcal{N}_i : 2 \leq |J| < |J_i|, && \text{(4c)} \\
& u_{i,\mathsf{t}} \leq v_{\mathsf{t}} && \forall\, \mathsf{t} \in \mathcal{T}_i, i \in [m], && \text{(4d)} \\
& \boldsymbol{v} \in \mathbb{B}^{|\mathcal{T}|}, \boldsymbol{u}_i \in \mathbb{B}^{|\mathcal{T}_i|} && \forall\, i \in [m]. &&
\end{aligned}
$$

3.2.1. Inequalities for monomials of degree 4 There are two minimum-sized linearization patterns for a multilinear term $x_j x_k x_l x_{l'}$: a) two linearizations of original variables followed by one linearization of artificial variables, e.g., $(\{j\}, \{k\}, \{j,k\})$, $(\{l\}, \{l'\}, \{l,l'\})$, and $(\{j,k\}, \{l,l'\}, \{j,k,l,l'\})$; or b) all linearizations involve at least one original variable, e.g.,

$(\{j\},\{k\},\{j,k\})$, $(\{j,k\},\{l\},\{j,k,l\})$, and $(\{j,k,l\},\{l'\},\{j,k,l,l'\})$. We explore the fact that these patterns use the same number of triples to derive inequalities that preserve at least one minimum-sized RML for any given MLP $f(\boldsymbol{x})$.

Isolated terms $x_jx_kx_l$ A multilinear term $x_jx_kx_l$ is *isolated* if it appears in exactly one monomial $\alpha_i f_i(\boldsymbol{x})$ and $f_i(\boldsymbol{x})$ has degree four, i.e., $f_i(\boldsymbol{x}) = x_jx_kx_lx_{l'}$ for some l' in $[n]$. By definition, the artificial variable $y_{\{j,k,l\}}$ can only be used to linearize $f_i(\boldsymbol{x})$. Therefore, an RML T that uses $y_{\{j,k,l\}}$ and $y_{\{j,k\}}$ can be replaced with another RML T' that uses $y_{\{j,k\}}$ and $y_{\{l,l'\}}$ instead. This operation does not increase the number of artificial variables, and if $y_{\{l,l'\}}$ appears in other monomials, we have $|T'| < |T|$; therefore, it follows that $|T'| \leq |T|$. We conclude that for every isolated term $x_jx_kx_l$ and set $\mathcal{T}_{j,k,l} := \{\mathsf{t} \in \mathcal{T} : \mathsf{head}(\mathsf{t}) = \{j,k,l\}\}$, there exists a minimum-sized RML that satisfies the equality below:

$$\sum_{\mathsf{t} \in \mathcal{T}_{j,k,l}} v_{\mathsf{t}} = 0. \tag{5}$$

Intermediate terms $x_jx_kx_l$ A multilinear term $x_jx_kx_l$ is *intermediate* if it only appears in monomials $\alpha_i f_i(\boldsymbol{x})$ of degree four (e.g., there is no monomial with term $f_{i'}(\boldsymbol{x}) = x_jx_kx_l$). In contrast to isolated terms, intermediate terms may appear in two or more monomials. Proposition 1 shows that there always exists one minimum-sized RML such that, for each triple t where $\mathsf{head}(\mathsf{t})$ is an intermediate term, either t is not used at all, or t is used in the linearization of at least two monomials of $f(\boldsymbol{x})$.

PROPOSITION 1. *For every triple* t *such that* $\mathsf{head}(\mathsf{t}) = \{j,k,l\}$ *is an intermediate term, the incorporation of the following inequality into* (4) *preserves at least one minimum-sized RML:*

$$2v_{\mathsf{t}} \leq \sum_{i \in [m]} u_{i,\mathsf{t}}. \tag{6}$$

Proof of Proposition 1: Let T be the proper triple set associated with a minimum-sized RML containing $\mathsf{t} = (\{j,k\},\{l\},\{j,k,l\})$ such that $x_jx_kx_l$ is an intermediate term and $\sum_{i\in[m]} u_{i,\mathsf{t}} = 1$, i.e., t is used to linearize exactly one term $f_i(\boldsymbol{x}) = x_jx_kx_lx_{l'}$. We show that T can be transformed into another RML that does not utilize t. As $|T|$ is minimum, $f_i(\boldsymbol{x})$ is linearized by t, $\mathsf{t}' = (\{j\},\{k\},\{j,k\})$, and $\mathsf{t}'' = (\{j,k,l\},\{l'\},\{j,k,l,l'\})$. As $x_jx_kx_l$ is an intermediate term and J_i is unique, i.e. $J_i \neq J_{i'}$ for $i' \in [m] \setminus \{i\}$, t'' can only be used in the linearization of $f_i(\boldsymbol{x})$. Therefore, t' is the only triple used to linearize $f_i(\boldsymbol{x})$ that may also be used to linearize other monomials. Thus, by replacing t with $(\{l\},\{l'\},\{l,l'\})$ and t'' with $(\{j,k\},\{l,l'\},\{j,k,l,l'\})$, we obtain an alternative proper triple set T' with the same cardinality as T and triples that can linearize $f_i(\boldsymbol{x})$. Repeating this procedure until all such triples t have been removed allows us to obtain a minimum-sized RML T^* that satisfies (6), so the result follows. ■

3.3. Hardness and Fixed-Parameter Tractability

This section introduces a fixed-parameter tractable algorithm for the 3-MLP, aspecial case of MLP in whici all monomials have degree at most 3. This result explores the similarities between the 3-MLP and the dominating set problem. Moreover, we show that finding a minimum-size RML is NP-hard; the identification of minimum-sized quadratization for pseudo-boolean functions is also NP-hard (Boros and Hammer (2002)), so this hardness result is expected.

3.3.1. Dominating Set Formulation of the 3-MLP Any RML of an MLP containing a monomial $f_i(\boldsymbol{x}) = x_j x_k x_l$ necessarily has a triple $\mathsf{t} = (\{j'\}, \{k'\}, \{j', k'\})$ for some $\{j', k'\} \subset \{j, k, l\}$, $j' \neq k'$, and one triple $\mathsf{t}' = (\{l'\}, \{j', k'\}, \{j', k', l'\})$, $l' \in \{j, k, l\} \setminus \{j', k'\}$. Therefore, we can cast an instance I' of the 3-MLP using a variation of the dominating set problem over a bipartite graph $G = (\mathbb{U}, \mathbb{V}, E)$. The *dominating set* of a graph consists of a subset V' of dominating vertices V such that any vertex in V is either in V' or has a neighbor in V'.

In our case, each vertex u of $\mathbb{U}$ is associated with a bilinear term $x_j x_k$ and an index set $J_u = \{j, k\}$, and each vertex v is associated with the multilinear term $x_l x_m x_n$ of a monomial $f_i(\boldsymbol{x})$ and an index set $J_v = \{l, m, n\}$; for ease of notation, we use $u = x_j x_k$ and $\mathsf{t} = x_l x_m x_n$. We adopt set-theoretical notation to represent the relationships between the elements of $\mathbb{U}$ and $\mathbb{V}$ based on their associated index sets (e.g., $u \cap v = \emptyset$ if $J_u \cap J_v = \emptyset$). Set E contains an edge $\{u, v\}$ if and only if $J_u \subset J_v$. For any $v \in \mathbb{V}$, we say that the vertices in $\mathbb{U}(v) := \{u \in \mathbb{U} : \{u, v\} \in E\}$ *cover* v; for example, if $v = x_i x_j x_k$, we have $U(v) = \{x_i x_j, x_i x_k, x_j x_k\}$. The identification of a minimum-size RML for the 3-MLP reduces to solving a special case of the dominating set problem on the graph G constructed as defined above, where *all the dominating vertices must be chosen from* $\mathbb{U}$. Next, we show that this problem (and the 3-MLP) is NP-hard.

3.3.2. Reduction Rules, Hardness, and Tractability We explore the connection with the dominating set problem to remove elements from $\mathbb{U}$ and $\mathbb{V}$ through a kernelization algorithm.

THEOREM 1 **(Reduction Rules)**. *The sequential, exhaustive, and iterative application of the following set of rules preserves at least one dominating set in G associated with a minimum RML:*

Rule 1 *For every $v \in \mathbb{V}$ such that $v \cap v' = \emptyset$ for every $v' \in \mathbb{V} \setminus \{v\}$, select an arbitrary pair $u \in \mathbb{U}(v)$ and remove v from $\mathbb{V}$ and all elements of $\mathbb{U}(v) \setminus \{u\}$ from $\mathbb{U}$.*

Rule 2 *Remove all elements of $\mathbb{U}$ of degree 1.*

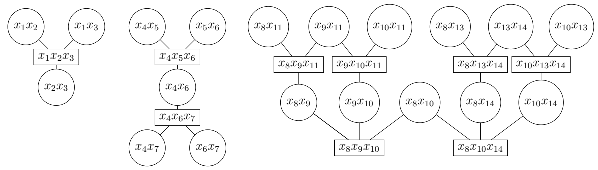
Rule 3 *For each element v of $\mathbb{V}$ with a single neighbor u, select u.*

Rule 4 *Remove all elements of $\mathbb{U}$ without neighbors.*

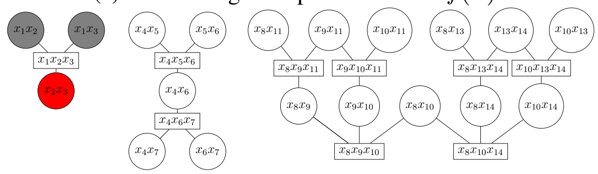
Rule 5 *The problem can be decomposed by its connected components in G.*

Proof: For **Rule 1**, observe that as v shares no variables with other triples in $\mathbb{V}$, all pairs in $\mathbb{U}(v)$ can only cover v. Therefore, any optimal solution of $f(\boldsymbol{x})$ has exactly one element of $\mathbb{U}(v)$. After the exhaustive application of **Rule 1**, each v has at least one neighbor u of degree at least two. Note that any optimal solution that uses a neighbor of v of degree one may be replaced for another solution of the same cardinality (or smaller) by using a neighbor of v of degree two instead. Therefore, we can remove all elements of $\mathbb{U}$ of degree one, i.e., we can apply **Rule 2**. The application of **Rule 1** and **Rule 2** may lead to configurations where an element v of $\mathbb{V}$ has only one neighbor in $\mathbb{U}$. As any feasible solution must contain at least one element of $\mathbb{U}(v)$ for each v in $\mathbb{V}$, we apply **Rule 3**. From the validity of the previous rules, it follows that there is at least one optimal solution that does not contain elements of $\mathbb{U}$ without neighbors, so **Rule 4** is valid. Finally, **Rule 5** follows from the fact that a vertex v of $\mathbb{V}$ cannot be covered by any element of $\mathbb{U}$ that does not belong to the same connected component in G. ■

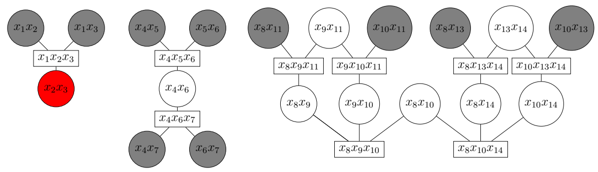
EXAMPLE 5. Figure 4 illustrates the application of the reduction rules on $f(\boldsymbol{x}) = x_1x_2x_3 + x_4x_5x_6 + x_4x_6x_7 + x_8x_9x_{10} + x_8x_9x_{11} + x_9x_{10}x_{11} + x_8x_{13}x_{14} + x_{10}x_{13}x_{14} + x_8x_{10}x_{14}$. The dominating set formulation of $f(\boldsymbol{x})$ is depicted in Figure 4a. Nodes of $\mathbb{U}$ incorporated into the optimal solution are shaded in red; eliminated nodes are shaded in gray. The term $x_1x_2x_3$ does not share variables with other terms, so we can apply **Rule 1** and select x_2x_3 to cover $x_1x_2x_3$ while excluding x_1x_2 and x_1x_3 (see Figure 4b). Next, **Rule 2** eliminates x_4x_5, x_5x_6, x_4x_7, x_6x_7, x_8x_{11}, $x_{10}x_{11}$, x_8x_{13}, and $x_{10}x_{13}$ (see Figure 4c). Finally, Figure 4d shows the result of **Rule 3**, where we select x_4x_6 to cover both $x_4x_5x_6$ and $x_4x_6x_7$.



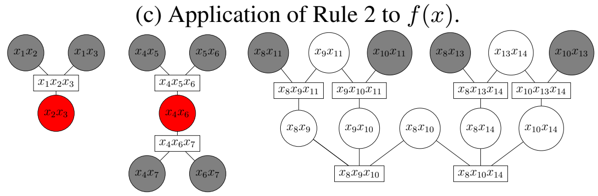
(a) Dominating set representation of $f(\boldsymbol{x})$.



(b) Application of Rule 1 to $f(x)$.



(c) Application of Rule 2 to $f(x)$.



(d) Application of Rule 3 to $f(x)$.

Figure 4 Reduction rules applied to $f(x) \coloneqq x_1x_2x_3 + x_4x_5x_6 + x_4x_6x_8 + x_7x_8x_9 + x_8x_9x_{10} + x_7x_9x_{10}$.

PROPOSITION 2 **(Structure of the Kernel).** *Let* $G^r = (\mathbb{U}^r, \mathbb{V}^r, E^r)$ *denote the graph resulting from the exhaustive application of the rules in Theorem 1.*

Property 1 *Each element of* $\mathbb{V}^r$ *has two or three neighbors in* $\mathbb{U}^r$.

Property 2 *Each non-selected element of* $\mathbb{U}^r$ *has at least two neighbors in* $\mathbb{V}^r$.

Property 3 G^r *is a* $K_{2,2}-$*free graph.*

Property 4 *If* $u \in \mathbb{U}^r$ *is not selected, any solution must contain at least one* $u' \in \mathbb{U}^r$ *for each neighbor of* u.

Proof: **Property 1** follows from the fact that $|\mathbb{U}(v)| = 3$ in G for any v in $\mathbb{V}^r$ and from **Rule 3**. **Property 2** follows directly from **Rule 2**. For **Property 3**, observe that any pair of elements u_1, u_2 in $\mathbb{U}$ sharing the same neighbors must have exactly one variable in common. Therefore, there are three variables associated with u_1 and u_2, so it defines exactly one element of $\mathbb{V}$, i.e., $\mathbb{V}$ cannot have two distinct elements that are simultaneously neighbors of both u_1 and u_2. Finally, **Property 4** follows directly from **Property 3**, as any vertex in $\mathbb{U}^r \setminus \{u\}$ can cover at most one neighbor of u. ■

Proposition 2 provides the conditions to adapt the arguments of Boros and Hammer (2002) to show that finding a minimum-size RML is NP-hard.

THEOREM 2. *Finding a minimum-size RML for the 3-MLP is NP-hard.*

Proof: The result follows from a reduction of the vertex cover problem, which is NP-hard (Garey and Johnson (1979)). In the vertex cover problem, we have a graph $G = (V, E)$ and wish to identify a subset V' of V such that, for each edge $e = \{u, v\}$ in E, we have $u \in V'$ or $v \in V'$ (or both). Let $G = (V, E)$ be the graph associated with an arbitrary instance I of the vertex cover

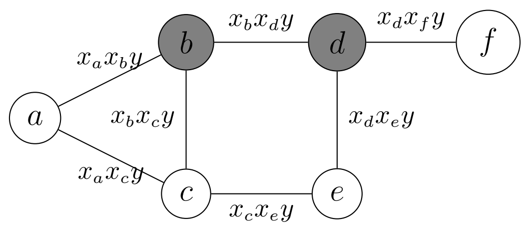


Figure 5 **Example of instance of the vertex cover problem for $G = (V, E)$, where $V = \{a, b, c, d, e, f\}$ and $E = \{\{a, b\}, \{a, c\}, \{b, c\}, \{b, d\}, \{c, e\}, \{d, e\}, \{d, f\}\}$. The figure shows the monomials associated with each edge; we have $\mathbb{V} = \{x_ax_by, x_ax_cy, x_bx_cy, x_bx_dy, x_cx_ey, x_dx_ey, x_dx_fy\}$. The 3-MLP instance is complete with $\mathbb{U} = \{x_ay, x_by, x_cy, x_dy, x_ey, x_fy\}$. An optimal solution for the vertex cover instance is the set $\{b, d\}$, whereas $\{x_by, x_dy\}$ is the optimal solution for the associated 3-MLP instance.**

problem. We construct the reduced bipartite graph $G^r = (\mathbb{U}^r, \mathbb{V}^r, E^r)$ associated with an instance of the 3-MLP. For each vertex v in V we have an element yx_v in $\mathbb{U}^r$, and for each edge $e = \{u, v\}$ in E we have an element x_ux_vy in $\mathbb{V}^r$. A complete construction would also require the inclusion of x_ux_v in $\mathbb{U}^r$ for each $\{u, v\}$ in E^r; however, it follows from **Rule 2** that we do not need to include them in $\mathbb{U}^r$, as there is at least one optimal solution of $(\mathbb{U}^r, \mathbb{V}^r)$ that does not use elements of $\mathbb{U}^r$ of degree 1. Therefore, we build E^r as in Section 3.3.1, but taking into account the transformations in Section 3.3.2. For an example, see Figure 5.

Any optimal solution V' for I is associated with a set of elements $\mathbb{U}'$ in $\mathbb{U}^r$ that covers each element of $\mathbb{V}^r$. In particular, the one-to-one relationships between V and $\mathbb{U}^r$ and E and $\mathbb{V}^r$ extend to the coverage of edges by vertices in G and triples by pairs in $(\mathbb{U}^r, \mathbb{V}^r)$. Therefore, it follows that the 3-MLP is NP-hard. ■

We leverage the construction used in the proof of Theorem 2 to show that `Greedy` may identify RMLs arbitrarily larger than a minimum-size RML.

PROPOSITION 3. *`Greedy` may identify RMLs arbitrarily larger than a minimum-size RML.*

Proof: Let $B = (U, V, E)$ be a bipartite graph such that $|U| = k$ for some $k \in \mathbb{N}$, and let $V := \bigcup_{i=1}^{k} V_i$ (i.e., V is partitioned into subsets $V_1, V_2, \ldots, V_k$) whereby $|V_i| = \lfloor \frac{k}{i} \rfloor$, $i \in [k]$. We construct E by assigning exactly i neighbors in U to each vertex in V_i, $i \in [k]$. Moreover, each vertex in U has at most one neighbor in V_i, and we assign neighbors in U to vertices in V_i so that the maximum degree of any vertex in U is $k-1$. We obtain an instance of the 3-MLP by applying the construction presented in the proof of Theorem 2 to graph B. The greedy algorithm proceeds by selecting, in each iteration, the vertex with the largest number of uncovered neighbors. By construction, all the $\sum_{i=1}^{k} |V_i|$ pairs associated with V are incorporated into the linearization by the greedy algorithm, so the solution size is $|V| \approx \sum_{i=1}^{k} \lfloor \frac{k}{i} \rfloor = \Theta(k \ln k)$. In contrast, this instance admits a linearization that picks all the pairs associated with U, which contains only k elements. Therefore, the proper triple set identified by `Greedy` is $O(\ln k)$ times larger than a minimum-sized one. ■

Finally, Theorem 3 shows that the 3-MLP is fixed-parameter tractable in the size of the linearization, i.e., for a fixed k, one can decide in polynomial time whether there exists a linearization with at most k elements.

THEOREM 3. *Given $k \in \mathbb{N}$, one can decide in time $O(k^6 + 3^k k^2)$ whether an instance of the 3-MLP has a linearization of size k.*

Proof: The structural properties of the reduced problem allow us to show that the 3-MLP is fixed-parameter tractable in the sizer k of the linearization; we denote this parameterized decision problem as (G^r, k). First, we show the adaptation of the kernelization procedure proposed by Buss and Goldsmith Buss and Goldsmith (1993) for the vertex cover problem applies to the 3-MLP.

LEMMA 1 **(Rule 6)**. *If G^r contains an element u in $\mathbb{U}$ with degree greater than or equal to $k+1$, remove u and its neighbors and solve $(G^r - u, k-1)$.*

Proof: This result follows from **Property 4**. Namely, if u has degree greater than or equal to $k+1$, then any solution for the 3-MLP that does not contain u must contain at least $k+1$ elements of $\mathbb{U} \setminus \{u\}$ to cover its neighborhood. Similarly, any certificate showing that $(G^r - u, k-1)$ is a "yes" instance can be efficiently converted in a "yes" certificate for (G^r, k). ■

Deleting a vertex u may affect all the elements in $\mathbb{V}$ and the elements in $\mathbb{U}$, so the application of Rule 6 takes time $O(|\mathbb{U}|(|\mathbb{U}| + |\mathbb{V}|))$. Our fixed-parameter tractable procedure to solve an instance (G^r, k) of the 3-MLP consists of applying Rules 1, 2, 3, 4, and 6; observe that, in addition to Rule 6, Rules 1 and 3 may also change (decrease) the value of k. We can omit Rule 5 for the decision version of the problem.

First, we claim that if (G^r, k) is a "yes" instance, then $|E| \leq k^2$. If Rule 6 (Proposition 1) cannot be applied, all vertices in $\mathbb{U}$ have at most k neighbors in $\mathbb{V}$. As at most k vertices of $\mathbb{U}$ may be selected and, consequently, at most k^2 vertices of $\mathbb{V}$ can be covered, it follows that $|E| \leq k^2$.

Next, we claim that if (G^r, k) is a "yes" instance, then $|\mathbb{V}| \leq k^2/2$ and $|\mathbb{U}| \leq k^2/2$. From **Property 1**, each element of $\mathbb{V}$ must have at least two neighbors in $\mathbb{U}$, so $|\mathbb{V}| \leq k^2/2$. Similarly, as **Property 2** shows that each element of $\mathbb{U}$ has at least two neighbors in $\mathbb{V}$, it follows that $|\mathbb{U}| \leq k^2/2$.

The exhaustive application of Rules 1, 2, 3, 4, and 6 can be performed in polynomial time. Namely, in each step, at least one vertex is removed, so in the worst case, we have $O((|\mathbb{U}| + |\mathbb{V}|)(2|\mathbb{U}| + |\mathbb{V}| + |\mathbb{V}|^2) + |\mathbb{U}|^2 + |\mathbb{U}||\mathbb{V}|) = O((|\mathbb{U}| + |\mathbb{V}|)(|\mathbb{U}|^2 + |\mathbb{V}|^2)) = O(w^3)$, where $w = |\mathbb{U}| + |\mathbb{V}|$ represents the size of the instance. A bounded search tree on the kernel needs time $T(w, k) = O(3^k n)$; each vertex in $\mathbb{V}$ has at most three neighbors, and a vertex of $\mathbb{U}$ can be removed (with its neighbors in $\mathbb{V}$) in time $O(w)$. As $w = O(k^2)$ after the kernelization procedure, the brute-force procedure consumes time $O(3^k k^2)$. In total, the algorithm consumes time $O(w^3 + 3^k k^2)$ = $O(k^6 + 3^k k^2)$; therefore, the 3-MLP is fixed-parameter tractable. ■

4. Best Bound LP Relaxation

Let $\widehat{\boldsymbol{v}} \in \mathbb{B}^{|\mathcal{T}|}$ be a binary vector representing a proper triple set T, i.e., $\hat{v}_{\mathsf{t}} = 1$ if and only if $\mathsf{t} \in T$. For a given $\widehat{\boldsymbol{v}} \in \mathbb{B}^{|\mathcal{T}|}$, the formulation presented in (2) can be rewritten as the following LP:

$$\min_{\boldsymbol{y} \in [0,1]^{|\mathcal{N}|}} \sum_{J \in \mathcal{N}} \beta_J y_J \tag{7a}$$

$$\text{s.t.} \underbrace{\begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}}_{=:B} \underbrace{\begin{pmatrix} y_J \\ y_{J'} \\ y_{J''} \end{pmatrix}}_{=:\boldsymbol{y}_{\mathsf{t}}} \leq \underbrace{\begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}}_{:=\mathbf{b}} + \underbrace{\begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}}_{=:\mathbf{c}} \hat{v}_{\mathsf{t}} \qquad \forall \mathsf{t} \in \mathcal{T} \tag{7b}$$

$$y_J \leq 1 \qquad \forall J \in \mathcal{N} \tag{7c}$$

We have $\beta_J = \alpha_i$ if $J = J_i$ for some $i \in [m]$ and $\beta_J = 0$ otherwise. Lemma 2 shows that the optimal solution to (7) is bounded for any choice of $\widehat{\boldsymbol{v}}$.

LEMMA 2. *The optimal objective value of* (7) *lies in the interval* $[\eta, 0]$ *where* $\eta = \sum_{i \in [m]} \min(0, \alpha_i)$.

Proof: The optimal objective value of (7) is not larger than zero since $\boldsymbol{y} = 0$ is feasible for any $\widehat{\boldsymbol{v}}$. A lower bound of $\sum_{J \in \mathcal{N}} \min(0, \beta_J)$ can be attained by setting $y_J = 1$ if $\beta_J < 0$ and $y_J = 0$ if $\beta_J \geq 0$. Since $\beta_J \neq 0$ only for $J \in \{J_1, \ldots, J_m\}$, the lower bound simplifies to $\sum_{i \in [m]} \min(0, \alpha_i)$. ■

Let $\boldsymbol{\lambda}_{\mathsf{t}} := (\lambda_{\mathsf{t},1}, \lambda_{\mathsf{t},2}, \lambda_{\mathsf{t},3})$ denote the vector with the dual multipliers for the inequalities in (7b) associated with the triple t. Let μ_J denote the multiplier for the bound constraint $y_J \leq 1$ in (7c) associated with the index set J. Moreover, let $\boldsymbol{\lambda}$ be a vector containing $\boldsymbol{\lambda}_{\mathsf{t}}$ for all $\mathsf{t} \in \mathcal{T}$, and $\boldsymbol{\mu}$ denote the collection of multipliers μ_J for all $J \in \mathcal{N}$. The dual of (7) can be written as follows:

$$\max_{(\boldsymbol{\lambda},\boldsymbol{\mu})\in\mathbb{R}^{|\mathcal{T}|}\times\mathbb{R}^{|\mathcal{N}|}} Obj(\boldsymbol{\lambda},\boldsymbol{\mu}) := -\sum_{\mathsf{t}\in\mathcal{T}}(\mathbf{b}^T\boldsymbol{\lambda}_{\mathsf{t}} + \mathbf{c}^T\boldsymbol{\lambda}_{\mathsf{t}}\widehat{v}_{\mathsf{t}}) - \sum_{J\in\mathcal{N}}\mu_J \tag{8a}$$

$$\text{s.t. } \beta_J + \sum_{\mathsf{t}:J=\mathsf{tail1(t)}}(-\lambda_{\mathsf{t},1}+\lambda_{\mathsf{t},3}) + \sum_{\mathsf{t}:J=\mathsf{tail2(t)}}(-\lambda_{\mathsf{t},2}+\lambda_{\mathsf{t},3})$$

$$+ \sum_{\mathsf{t}:J=\mathsf{head(t)}}(\lambda_{\mathsf{t},1}+\lambda_{\mathsf{t},2}-\lambda_{\mathsf{t},3}) + \mu_J \geq 0 \qquad \forall\, J\in\mathcal{N} \tag{8b}$$

$$\boldsymbol{\lambda},\boldsymbol{\mu} \geq 0 \tag{8c}$$

From Lemma 2 and strong duality, it follows that the optimal solution to (8) is bounded given $\widehat{\boldsymbol{v}}$.

LEMMA 3. *The optimal objective value of* (8) *lies in the interval* $[\eta, 0]$.

4.1. Bounds on the dual multipliers

In this section, we prove that, for any given $\widehat{\boldsymbol{v}}$, the multipliers $\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}$ and $\boldsymbol{\mu}^{\widehat{\boldsymbol{v}}}$ have finite bounds independent of $\widehat{\boldsymbol{v}}$; this result is stated in Proposition 4 and follows from Lemmas 4, 5, and 6. For brevity, we assume in this section that $\widehat{\boldsymbol{v}}$ is fixed, so we omit it from the notation.

PROPOSITION 4. *Let $\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}, \boldsymbol{\mu}^{\widehat{\boldsymbol{v}}}$ denote an optimal solution to* (8) *for a fixed $\widehat{\boldsymbol{v}}$. We have $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},j} \leq M_{\mathsf{t},j}$ for all $\mathsf{t}\in\mathcal{T}$, $j = 1,2,3$ with $M_{\mathsf{t},j}\in[0,\infty)$ and $\mu^{\widehat{\boldsymbol{v}}}_J \leq M_J$ for all $J\in\mathcal{N}$ with $M_J\in[0,\infty)$.*

LEMMA 4. *There exists an optimal solution $\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}, \boldsymbol{\mu}^{\widehat{\boldsymbol{v}}}$ to* (8) *with $\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}_{\mathsf{t}} = 0$ for all t such that $\widehat{v}_{\mathsf{t}} = 0$.*

Proof: Suppose $(\widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}})$ is feasible for (8) and $\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}'} \neq 0$ for some t' such that $\widehat{v}_{\mathsf{t}'} = 0$. The result follows from the fact that there exists a solution $(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}})$ that is feasible for (8) with $\widehat{\boldsymbol{\lambda}}_{\mathsf{t}'} = 0$ and $Obj(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}}) = Obj(\widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}})$. Let $\mathsf{t}' = (J', J'', J''')$. Define $\widehat{\boldsymbol{\lambda}}$ and $\widehat{\boldsymbol{\mu}}$ as follows:

$$\widehat{\boldsymbol{\lambda}}_{\mathsf{t}} = \begin{cases} \widetilde{\boldsymbol{\lambda}}_{\mathsf{t}} \text{ if } \mathsf{t}\neq\mathsf{t}' \\ 0 \text{ if } \mathsf{t}=\mathsf{t}' \end{cases} \text{ and } \widehat{\mu}_J = \begin{cases} \widetilde{\mu}_J \text{ if } J\notin\{J',J'',J'''\} \\ \widetilde{\mu}_{J'} + \widetilde{\lambda}_{\mathsf{t}',3} \text{ if } J = J' \\ \widetilde{\mu}_{J''} + \widetilde{\lambda}_{\mathsf{t}',3} \text{ if } J = J'' \\ \widetilde{\mu}_{J'''} + \widetilde{\lambda}_{\mathsf{t}',1} + \widetilde{\lambda}_{\mathsf{t}',2} \text{ if } J = J'''. \end{cases} \tag{9}$$

By construction, we have $\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}} \geq 0$. Moreover, $\widehat{\boldsymbol{\lambda}}_{\mathsf{t}'}$, $\widehat{\boldsymbol{\mu}}_{J'}$, $\widehat{\boldsymbol{\mu}}_{J''}$, and $\widehat{\boldsymbol{\mu}}_{J'''}$ only figure in the inequalities in (8b) for $J\in\{J',J'',J'''\}$. Hence, the inequality (8b) holds for all $J\setminus\{J',J'',J'''\}$. Therefore, we just need to show that $(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}})$ satisfy (8b) for $\{J',J'',J'''\}$.

First, consider the left-hand side of (8b) for $J = J'$; the analysis for $J = J''$ is identical. We have

$$\beta_{J'} + \sum_{\mathsf{t}:J'=\mathsf{tail1(t)}}(-\widehat{\lambda}_{\mathsf{t},1}+\widehat{\lambda}_{\mathsf{t},3}) + \sum_{\mathsf{t}:J'=\mathsf{tail2(t)}}(-\widehat{\lambda}_{\mathsf{t},2}+\widehat{\lambda}_{\mathsf{t},3})$$

$$+ \sum_{\mathsf{t}:J'=\mathsf{head(t)}}(\widehat{\lambda}_{\mathsf{t},1}+\widehat{\lambda}_{\mathsf{t},2}-\widehat{\lambda}_{\mathsf{t},3}) + \widehat{\mu}_{J'} \tag{10a}$$

$$= \beta_{J'} + (-\widehat{\lambda}_{\mathsf{t}',1}+\widehat{\lambda}_{\mathsf{t}',3}) + \sum_{\mathsf{t}:J'=\mathsf{tail1(t)}\setminus\{\mathsf{t}'\}}(-\widehat{\lambda}_{\mathsf{t},1}+\widehat{\lambda}_{\mathsf{t},3}) + \sum_{\mathsf{t}:J'=\mathsf{tail2(t)}}(-\widehat{\lambda}_{\mathsf{t},2}+\widehat{\lambda}_{\mathsf{t},3})$$

$$+\sum_{\mathsf{t}:J'=\mathsf{head(t)}} (\widehat{\lambda}_{\mathsf{t},1}+\widehat{\lambda}_{\mathsf{t},2}-\widehat{\lambda}_{\mathsf{t},3})+\widehat{\mu}_{J'} \tag{10b}$$

$$=\beta_{J'}+0+\sum_{\mathsf{t}:J'=\mathsf{tail1(t)}\setminus\{\mathsf{t}'\}} (-\widetilde{\lambda}_{\mathsf{t},1}+\widetilde{\lambda}_{\mathsf{t},3})+\sum_{\mathsf{t}:J'=\mathsf{tail2(t)}} (-\widetilde{\lambda}_{\mathsf{t},2}+\widetilde{\lambda}_{\mathsf{t},3})$$
$$+\sum_{\mathsf{t}:J'=\mathsf{head(t)}} (\widetilde{\lambda}_{\mathsf{t},1}+\widetilde{\lambda}_{\mathsf{t},2}-\widetilde{\lambda}_{\mathsf{t},3})+\widetilde{\mu}_{J'}+\widetilde{\lambda}_{\mathsf{t}',3} \tag{10c}$$

$$=\beta_{J'}+\sum_{\mathsf{t}:J'=\mathsf{tail1(t)}} (-\widetilde{\lambda}_{\mathsf{t},1}+\widetilde{\lambda}_{\mathsf{t},3})+\sum_{\mathsf{t}:J'=\mathsf{tail2(t)}} (-\widetilde{\lambda}_{\mathsf{t},2}+\widetilde{\lambda}_{\mathsf{t},3})$$
$$+\sum_{\mathsf{t}:J'=\mathsf{head(t)}} (\widetilde{\lambda}_{\mathsf{t},1}+\widetilde{\lambda}_{\mathsf{t},2}-\widetilde{\lambda}_{\mathsf{t},3})+\widetilde{\mu}_{J'}+\widetilde{\lambda}_{\mathsf{t}',1}\geq\widetilde{\lambda}_{\mathsf{t}',1}\geq 0 \tag{10d}$$

In the first equality, we move the terms associated with t' out of the first summation. The equality in (10c) is obtained by substituting (9), and (10d) follows by adding and subtracting the term $\widetilde{\lambda}_{\mathsf{t}',1}$ and collecting the term $(-\widetilde{\lambda}_{\mathsf{t}',1}+\widetilde{\lambda}_{\mathsf{t}',3})$ into the first summation. The first inequality is obtained from (8b) holding for $(\widetilde{\boldsymbol{\lambda}},\widetilde{\boldsymbol{\mu}})$, and the final inequality follows from $\widetilde{\boldsymbol{\lambda}}\geq 0$. For $J=J'''$, we have

$$\beta_{J'''}+\sum_{\mathsf{t}:J'''=\mathsf{tail1(t)}} (-\widehat{\lambda}_{\mathsf{t},1}+\widehat{\lambda}_{\mathsf{t},3})+\sum_{\mathsf{t}:J'''=\mathsf{tail2(t)}} (-\widehat{\lambda}_{\mathsf{t},2}+\widehat{\lambda}_{\mathsf{t},3})$$
$$+\sum_{\mathsf{t}:J'''=\mathsf{head(t)}} (\widehat{\lambda}_{\mathsf{t},1}+\widehat{\lambda}_{\mathsf{t},2}-\widehat{\lambda}_{\mathsf{t},3})+\widehat{\mu}_{J'''} \tag{11a}$$

$$=\beta_{J'''}+\sum_{\mathsf{t}:J'''=\mathsf{tail1(t)}} (-\widehat{\lambda}_{\mathsf{t},1}+\widehat{\lambda}_{\mathsf{t},3})+\sum_{\mathsf{t}:J'''=\mathsf{tail2(t)}} (-\widehat{\lambda}_{\mathsf{t},2}+\widehat{\lambda}_{\mathsf{t},3})$$
$$+(\widehat{\lambda}_{\mathsf{t}',1}+\widehat{\lambda}_{\mathsf{t}',2}-\widehat{\lambda}_{\mathsf{t}',3})+\sum_{\mathsf{t}:J'''=\mathsf{head(t)}\setminus\{\mathsf{t}'\}} (\widehat{\lambda}_{\mathsf{t},1}+\widehat{\lambda}_{\mathsf{t},2}-\widehat{\lambda}_{\mathsf{t},3})+\widehat{\mu}_{J'''} \tag{11b}$$

$$=\beta_{J'''}+\sum_{\mathsf{t}:J'''=\mathsf{tail1(t)}} (-\widetilde{\lambda}_{\mathsf{t},1}+\widetilde{\lambda}_{\mathsf{t},3})+\sum_{\mathsf{t}:J'''=\mathsf{tail2(t)}} (-\widetilde{\lambda}_{\mathsf{t},2}+\widetilde{\lambda}_{\mathsf{t},3})$$
$$+\sum_{\mathsf{t}:J'''=\mathsf{head(t)}\setminus\{\mathsf{t}'\}} (\widetilde{\lambda}_{\mathsf{t},1}+\widetilde{\lambda}_{\mathsf{t},2}-\widetilde{\lambda}_{\mathsf{t},3})+0+\widetilde{\mu}_{J'''}+\widetilde{\lambda}_{\mathsf{t}',1}+\widetilde{\lambda}_{\mathsf{t}',2} \tag{11c}$$

$$=\beta_{J'''}+\sum_{\mathsf{t}:J'''=\mathsf{tail1(t)}} (-\widetilde{\lambda}_{\mathsf{t},1}+\widetilde{\lambda}_{\mathsf{t},3})+\sum_{\mathsf{t}:J'''=\mathsf{tail2(t)}} (-\widetilde{\lambda}_{\mathsf{t},2}+\widetilde{\lambda}_{\mathsf{t},3})$$
$$+\sum_{\mathsf{t}:J'''=\mathsf{head(t)}} (\widetilde{\lambda}_{\mathsf{t},1}+\widetilde{\lambda}_{\mathsf{t},2}-\widetilde{\lambda}_{\mathsf{t},3})+\widetilde{\mu}_{J'''}+\widetilde{\lambda}_{\mathsf{t}',3}\geq\widetilde{\lambda}_{\mathsf{t}',3}\geq 0 \tag{11d}$$

Equality (11b) follows from splitting the last summation over t'. The equality in (11c) is obtained by substituting (9), and (11d) follows by adding and subtracting the term $\widetilde{\lambda}_{\mathsf{t}',3}$ and collecting the term $(\widetilde{\lambda}_{\mathsf{t}',1}+\widetilde{\lambda}_{\mathsf{t}',2}-\widetilde{\lambda}_{\mathsf{t}',3})$ into the last summation. The first inequality is obtained from (8b) holding for the variables $(\widetilde{\boldsymbol{\lambda}},\widetilde{\boldsymbol{\mu}})$, and the final inequality follows from $\widetilde{\boldsymbol{\lambda}}\geq 0$. Therefore, $(\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\mu}})$ is feasible for (8b) and $\boldsymbol{\lambda}_{\mathsf{t}'}=0$. Finally, we show that $Obj(\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\mu}})=Obj(\widetilde{\boldsymbol{\lambda}},\widetilde{\boldsymbol{\mu}})$. This follows from:

$$Obj(\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\mu}})=-\sum_{\mathsf{t}\in\mathcal{T}}(b^T\widehat{\boldsymbol{\lambda}}_{\mathsf{t}}+c^T\widehat{\boldsymbol{\lambda}}_{\mathsf{t}}\widehat{v}_{\mathsf{t}})-\sum_{J\in\mathcal{N}}\widehat{\mu}_J \tag{12a}$$

$$=-\sum_{\mathsf{t}\in\mathcal{T}\setminus\{\mathsf{t}'\}}(b^T\widehat{\boldsymbol{\lambda}}_{\mathsf{t}}+c^T\widehat{\boldsymbol{\lambda}}_{\mathsf{t}}\widehat{v}_{\mathsf{t}})-(b^T\widehat{\boldsymbol{\lambda}}_{\mathsf{t}'}+c^T\widehat{\boldsymbol{\lambda}}_{\mathsf{t}'}\widehat{v}_{\mathsf{t}'})$$

$$
\begin{aligned}
&- \sum_{J\in\mathcal{N}\setminus\{J',J'',J'''\}} \widehat{\mu}_J - \widehat{\mu}_{J'} - \widehat{\mu}_{J''} - \widehat{\mu}_{J'''} && \text{(12b)}\\
&= - \sum_{\mathsf{t}\in\mathcal{T}\setminus\{\mathsf{t}'\}} (b^T\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}} + c^T\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}}\widehat{v}_{\mathsf{t}}) - 0 \\
&- \sum_{J\in\mathcal{N}\setminus\{J',J'',J'''\}} \widetilde{\mu}_J - \widetilde{\mu}_{J'} - \widetilde{\mu}_{J''} - \widetilde{\mu}_{J'''} - \widetilde{\lambda}_{\mathsf{t}',1} - \widetilde{\lambda}_{\mathsf{t}',2} - 2\widetilde{\lambda}_{\mathsf{t}',3} && \text{(12c)}\\
&= - \sum_{\mathsf{t}\in\mathcal{T}\setminus\{\mathsf{t}'\}} (b^T\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}} + c^T\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}}\widehat{v}_{\mathsf{t}}) - \sum_{J\in\mathcal{N}} \widetilde{\mu}_J - b^T\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}'} = Obj(\widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}}) && \text{(12d)}
\end{aligned}
$$

Equality (12b) follows by splitting the first sum in t' and the second sum in $\{J', J'', J'''\}$. The equality in (12c) follows by substituting (9). The equality in (12d) follows from the definition of b in (7b). The final equality follows by noting that $b^T\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}'} = b^T\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}'} + c^T\widetilde{\boldsymbol{\lambda}}_{\mathsf{t}'}v_{\mathsf{t}'}$ since $v_{\mathsf{t}'} = 0$ and collecting the terms into the summation over t in $\mathcal{T}\setminus\{\mathsf{t}'\}$. ■

LEMMA 5. *Let* $(\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}, \boldsymbol{\mu}^{\widehat{\boldsymbol{v}}})$ *be an optimal solution to* (8) *as stated in Lemma 4. Then* $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3}, \mu^{\widehat{\boldsymbol{v}}}_J \le \eta$.

Proof: Consider the term $\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}_{\mathsf{t}}$ in (8a) for some triple t. This can be simplified as

$$
\mathbf{b}^T\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}_{\mathsf{t}} + \mathbf{c}^T\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}_{\mathsf{t}}\widehat{v}_{\mathsf{t}} = \begin{cases} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} \text{ if } \widehat{v}_{\mathsf{t}} = 1; \\ \mathbf{b}^T\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}_{\mathsf{t}} = 0 \text{ if } \widehat{v}_{\mathsf{t}} = 0, \end{cases} \tag{13}
$$

which follows by substituting for $\mathbf{b}, \mathbf{c}$ from (7b) and from Lemma 4. Thus, the optimal value of the objective in (8a) can be reduced to

$$
Obj(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}}) = -\sum_{\mathsf{t}\in\mathcal{T}} (\mathbf{b}^T\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}_{\mathsf{t}} + \mathbf{c}^T\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}_{\mathsf{t}}\widehat{v}_{\mathsf{t}}) - \sum_{J\in\mathcal{N}} \mu^{\widehat{\boldsymbol{v}}}_J = -\sum_{\mathsf{t}\in\mathcal{T}:\widehat{v}_{\mathsf{t}}=1} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} - \sum_{J\in\mathcal{N}} \mu^{\widehat{\boldsymbol{v}}}_J \ge -\eta, \tag{14}
$$

where the first equality follows from (13) and the inequality from Lemma 3. Combining $\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}, \boldsymbol{\mu}^{\widehat{\boldsymbol{v}}} \ge 0$ with (14) yields that $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} \le \eta$ for all t in $\mathcal{T}$ such that $\widehat{v}_{\mathsf{t}} = 1$ and $\mu^{\widehat{\boldsymbol{v}}}_J \le \eta$ for all J in $\mathcal{N}$. To complete the proof it suffices to recall that, by Lemma 4, $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} = 0 \le \eta$ for all t in $\mathcal{T}$ such that $\widehat{v}_{\mathsf{t}} = 0$. ■

Lemma 5 yields that $M_{\mathsf{t},3} = \eta$ for all t in $\mathcal{T}$ and $M_J = \eta$ for all J in $\mathcal{N}$. Next, Lemma 6 shows that the bounds for $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1}$ and $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2}$ are also finite for all t in $\mathcal{T}$.

LEMMA 6. *Let* $(\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}, \boldsymbol{\mu}^{\widehat{\boldsymbol{v}}})$ *be an optimal solution to* (8) *as stated in Lemma 4. There exists a finite* $M_{\mathsf{t},j}$ *for each* $\mathsf{t}\in\mathcal{T}$ *and* $j = 1, 2$ *such that* $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},j} \le M_{\mathsf{t},j}$.

Proof: Consider the inequality in (8b) for $J \in \mathcal{N}$. This can be rewritten for $(\boldsymbol{\lambda}^{\widehat{\boldsymbol{v}}}, \boldsymbol{\mu}^{\widehat{\boldsymbol{v}}})$ as

$$
\begin{aligned}
&\sum_{\mathsf{t}:J=\mathsf{tail1}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1} + \sum_{\mathsf{t}:J=\mathsf{tail2}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2} \le \\
&\beta_J + \sum_{\mathsf{t}:J=\mathsf{tail1}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} + \sum_{\mathsf{t}:J=\mathsf{tail2}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} + \sum_{\mathsf{t}:J=\mathsf{head}(\mathsf{t})} (\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1} + \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2} - \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3}) + \mu^{\widehat{\boldsymbol{v}}}_J
\end{aligned} \tag{15}
$$

From (14) we have that $\sum_{\mathsf{t}\in\mathcal{T}} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} + \sum_{J\in\mathcal{N}} \mu^{\widehat{\boldsymbol{v}}}_J \le \eta$. Then we can upper bound the terms involving $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3}$ and $\mu^{\widehat{\boldsymbol{v}}}_J$ on the right-hand side of (15) as

$$
\sum_{\mathsf{t}:J=\mathsf{tail1}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} + \sum_{\mathsf{t}:J=\mathsf{tail2}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} + \mu^{\widehat{\boldsymbol{v}}}_J \le \sum_{\mathsf{t}:\mathcal{T}} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3} + \sum_{J\in\mathcal{N}} \mu^{\widehat{\boldsymbol{v}}}_J \le \eta \tag{16}
$$

where the first inequality follows by noting that either $J = \mathsf{tail1}(\mathsf{t})$ or $J = \mathsf{tail2}(\mathsf{t})$ but not both, and from the non-negativity of multipliers. The second inequality follows from (14) and Lemma 4. Thus, the inequality (15) is equivalent to

$$\begin{aligned}\sum_{\mathsf{t}:J=\mathsf{tail1}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1} + \sum_{\mathsf{t}:J=\mathsf{tail2}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2} &\le \beta_J + \eta + \sum_{\mathsf{t}:J=\mathsf{head}(\mathsf{t})} (\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1} + \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2} - \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3}) \\ &\le \beta_J + \eta + \sum_{\mathsf{t}:J=\mathsf{head}(\mathsf{t})} (\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1} + \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2}) \end{aligned} \tag{17}$$

where the first inequality follows from (16) and the second from the non-negativity of $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},3}$. Observe that the right-hand side of (17) involves the multipliers $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1}$ and $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2}$ for all t such that $J = \mathsf{head}(\mathsf{t})$, i.e., the triples t for which J is the head. If an upper bound is available for such multipliers, then we can use (17) to derive an upper bound on the arcs in which J is a tail.

We show by induction that $M_{\mathsf{t},1}$ and $M_{\mathsf{t},2}$ are finite. First, consider $\mathcal{J}_1 := \{J \in \mathcal{N} \mid |J| = 1\}$. We have $\{\mathsf{t} \in \mathcal{T} \mid J = \mathsf{head}(\mathsf{t})\} = \emptyset$ for each J in $\mathcal{J}_1$, i.e., J cannot be the head of any triple. Consequently, the inequality (17) for $J \in \mathcal{J}_1$ becomes

$$\sum_{\mathsf{t}:J=\mathsf{tail1}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1} + \sum_{\mathsf{t}:J=\mathsf{tail2}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2} \le \beta_J + \eta. \tag{18}$$

Therefore, $M_{\mathsf{t},1} \le \beta_J + \eta$ and $M_{\mathsf{t},2} \le \beta_J + \eta$ for each t in $\mathcal{T}$ such that $\mathsf{tail1}(\mathsf{t}) \in \mathcal{J}_1$ or $\mathsf{tail2}(\mathsf{t}) \in \mathcal{J}_1$, respectively. Next, we consider $\mathcal{J}_2 := \{J \in \mathcal{N} \mid |J| = 2\}$. For each J in $\mathcal{J}_2$, any t in $\{\mathsf{t} \in \mathcal{T} \mid J = \mathsf{head}(\mathsf{t})\}$ is such that $|\mathsf{tail1}(\mathsf{t})| = 1$ and $|\mathsf{tail2}(\mathsf{t})| = 1$. Therefore, upper bounds $M_{\mathsf{t},1}$ and $M_{\mathsf{t},2}$ have been identified for $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1}$ and $\lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2}$, respectively, in the first iteration. Hence (17) can be written as

$$\sum_{\mathsf{t}:J=\mathsf{tail1}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},1} + \sum_{\mathsf{t}:J=\mathsf{tail2}(\mathsf{t})} \lambda^{\widehat{\boldsymbol{v}}}_{\mathsf{t},2} \le \beta_J + \eta + \sum_{\mathsf{t}:J=\mathsf{head}(\mathsf{t})} (M_{\mathsf{t},1} + M_{\mathsf{t},2}). \tag{19}$$

Thus $M_{\mathsf{t},1}$ for $\mathsf{tail1}(\mathsf{t}) \in \mathcal{J}_2$ and $M_{\mathsf{t},2}$ for $\mathsf{tail2}(\mathsf{t}) \in \mathcal{J}_2$ can be obtained from the right-hand side of (19). We can repeat the above for $\mathcal{J}_k := \{J \in \mathcal{N} || J| = k\}$, $3 \le k \le n$, by considering sets of increasing cardinality to determine all the bounds $M_{\mathsf{t},j}$ for $j = 1, 2$. ■

4.2. Best Bound MIP

Let $\boldsymbol{\mathcal{V}}$ denote the set of vectors $\boldsymbol{v}$ in $\mathbb{B}^{|\mathcal{T}|}$ composing a feasible solution to (4), and let $\boldsymbol{\mathcal{V}}_k := \{\boldsymbol{v} | \boldsymbol{v} \in \boldsymbol{\mathcal{V}}, \|\boldsymbol{v}\|_1 \le k\}$ be the collection of proper triple sets containing at most k elements. We consider the following bilevel formulation to identify an element of $\boldsymbol{\mathcal{V}}_k$ that yields the best LP relaxation bound.

$$\max_{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}_k} \min_{\boldsymbol{y} \in [0,1]^{|\mathcal{N}|}} \sum_{i=1}^{m} \alpha_i y_{J_i} \tag{20a}$$

$$\text{s.t. } B\boldsymbol{y}_{\mathsf{t}} \le \mathbf{b} + \mathbf{c} v_{\mathsf{t}} \qquad \forall\, \mathsf{t} = (J, J', J'') \in \mathcal{T} \tag{20b}$$

Variables $\boldsymbol{y}$ and $\boldsymbol{v}$ are defined over $\mathcal{N}$ and $\mathcal{T}$, respectively, as in (7). Furthermore, we use $\boldsymbol{y}_{\mathsf{t}}$ to denote the collection of variables $(y_J, y_{J'}, y_{J''})$, where $\mathsf{t} = (J, J', J'')$. Finally, we use strong duality to cast (20) as a single-level maximization MIP.

THEOREM 4. *The max-min problem in* (20) *is equivalent to the following MIP:*

$$\max_{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}_K, \boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{T}|}, \boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{N}|}} \; -\sum_{\mathsf{t} \in \mathcal{T}} \lambda_{\mathsf{t},3} - \sum_{J \in \mathcal{N}} \mu_J \tag{21}$$

$$\textit{s.t.} \qquad (8b) - (8c) \tag{22}$$

$$\lambda_{\mathsf{t},j} \le M_{\mathsf{t},j} v_{\mathsf{t}} \qquad \mathsf{t} \in \mathcal{T}, j = 1, 2, 3. \tag{23}$$

Proof: First, we show that (20) can be cast as the following single-level problem:

$$\max_{\boldsymbol{v}\in\boldsymbol{\mathcal{V}}_k,\boldsymbol{\lambda}\in\mathbb{R}^{|\mathcal{T}|},\boldsymbol{\mu}\in\mathbb{R}^{|\mathcal{N}|}} \; -\sum_{\mathrm{t}\in\mathcal{T}}(\mathbf{b}^T\boldsymbol{\lambda}_{\mathrm{t}}+\mathbf{c}^T\boldsymbol{\lambda}_{\mathrm{t}}v_{\mathrm{t}})-\sum_{J\in\mathcal{N}}\mu_J \tag{24a}$$

$$\text{s.t. (8b)} - \text{(8c)} \tag{24b}$$

From Lemma 2 we have that the inner minimization problem in (20), given by (7), attains a finite optimal value for any $\boldsymbol{v}$. By strong duality of LP, the optimal objective value of the inner minimization problem equals the optimal objective value of the dual (8). Substituting (7) by (8) and noting that $\max_{\boldsymbol{v}\in\boldsymbol{\mathcal{V}}_k}$ and $\max_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ can be combined into a single level proves the claim. Formulation (24) has linear constraints but a bilinear objective, since v_{t} multiplies $\mathbf{c}^T\lambda_{\mathrm{t}}$. By Lemma 4, the optimal solution to (8) satisfies $\boldsymbol{\lambda}_{\mathrm{t}}^{\boldsymbol{v}}=0$ for each t such that $v_{\mathrm{t}}=0$. Lemmas 5 and 6 provide upper bounds on the optimal multipliers $(\boldsymbol{\lambda}^{\widehat{v}},\boldsymbol{\mu}^{\widehat{v}})$. Hence, the constraints (23) are valid. Finally, the simplification of the objective function follows from (14). ■

5. Computational results

This section presents the results of a numerical study conducted to demonstrate the benefits of the minimum-size and best-bound linearization strategies introduced in §3.2 and §4.2. We denote by `MinLin` the minimum-size linearization derived from the MIP (4) with inequalities (5) and (6). Similarly, we use `BB` to denote the best-bound linearization obtained from the solution of the MIP presented in Theorem 4.

We start in §5.1 by describing the test set. In §5.2 and §5.3, we compare different linearization strategies in terms of the number of variables and the bounds of the resulting LP relaxations. Then, in §5.4 and §5.5, we compare `MinLin` and `BB` with the algorithms `EV-MinLin` and `EV-BB`, proposed by Elloumi and Verchère (2023) for identifying minimum-size and best-bound linearizations, respectively. Finally, in §5.6, we use different linearizations to reformulate the MLPs of our test set as QCPs, which we then solve with the global solver GUROBI (Gurobi Optimization, LLC (2022)). This approach is compared with a different strategy where we directly solve the original MLPs with the global solver BARON (Khajavirad and Sahinidis (2018)).

Our code was implemented in Python 3.10. We use Pyomo as modeling language (Hart et al. (2017)) and run our experiments on an Apple M1 Pro with 10 cores and 32 GB of RAM. We use GUROBI 11.0.2 to solve all the MIPs and QCPs of our experiments. To solve the MLPs, we use BARON 24.5.8, with CPLEX 22.1.1 (Cplex (2022)) as an LP subsolver. We use default settings for all solvers. The time spent with the construction of the models is excluded from all experiments. For the comparisons with `EV-MinLin` and `EV-BB`, we use the original Julia implementations of these algorithms, which were kindly provided by the authors. Our data and code are available on GitHub (Cardonha et al. (2024)).

We use plots with performance profiles to report the computational performance of the algorithms. These plots are divided into two parts. On the left, we report the number of instances solved to optimality (in the y-axis) within the amount of time indicated in the x-axis; the largest value of x is the time limit. On the right, we indicate the number of instances for which the solver obtained an optimality gap inferior to the value indicated in the x-axis within the time limit. Whenever applicable, we omit and scale parts of the plots to make performance differences more clear. We draw a horizontal line indicating the total number of instances in situations where no algorithm obtains non-trivial bounds for all instances. We occasionally omit from our plots trivial instances, i.e., instances that can be solved by all algorithms in less than one second. When we solve MIPs, QCPs or MLPs to global optimality, we calculate the optimality gaps as

$$\text{Optimality Gap} = \left(\frac{|f_{\texttt{ub}} - f_{\texttt{lb}}|}{|f_{\texttt{ub}}| + 10^{-9}} \right) \times 100.$$

where $f_{\texttt{ub}}$ and $f_{\texttt{lb}}$ respectively denote the best upper and lower bounds obtained within the time limit.

5.1. Test set

In the following, we describe the three families of instances considered in our experiments.

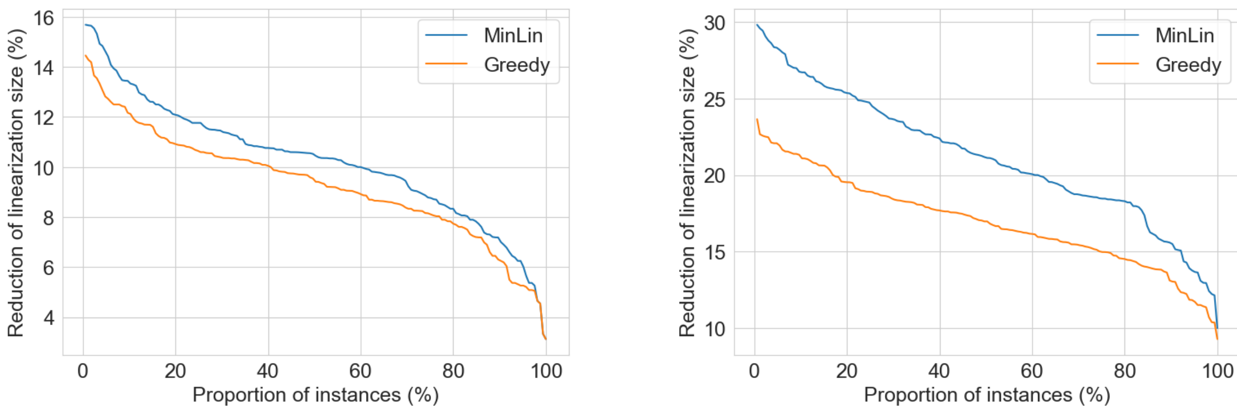
Multilinear Optimization Problems: This collection consists of 330 unconstrained multilinear problems with continuous variables, with $\Omega = [0,1]^n$. We created these instances by following the procedure that Del Pia et al. (2020) used to generate their random instances. Namely, for each combination of $n \in \{20, 25, 30, 35, 40\}$ and $m \in \{50, 60, \ldots, 150\}$, we generated three instances in which all monomials have degree 3 and three instances in which all monomials have degree 4. The variables of each monomial were chosen independently and uniformly at random. The coefficients of each monomial are integer values chosen from a uniform distribution defined over the interval $[-100, 100]$. The resulting collection contains 165 instances of degree 3 and 165 instances of degree 4, which we respectively denote by `mult3` and `mult4`.

Vision Instances: The `vision` instances represent an image restoration problem in which the objective function is given by $f(\boldsymbol{x}) = L(\boldsymbol{x}) + H(\boldsymbol{x})$, where $L(\boldsymbol{x})$ is an affine function and $H(\boldsymbol{x})$ a multilinear function of degree four. In this case, all variables are binary. For our experiments, we used the 45 instances generated by Crama and Rodríguez-Heck (2017), for which $n \in \{100, 150, 225\}$. The instances of a given size share the same multilinear function $H(\boldsymbol{x})$, i.e., they only differ in the coefficients of $L(\boldsymbol{x})$.

Auto-correlation Instances: The `autocorr` instances also have 0-1 variables and were extracted from POLIP, a library of polynomially-constrained mixed-integer programming problems (`http://polip.zib.de`). We consider 33 instances with $n \in \{20, 25, 30, 35, 40, 45, 50\}$ in our experiments.

5.2. Linearization size

For each instance of the test set, we constructed 3 different RMLs by using the linearization strategies `Seq`, `Greedy` and `MinLin`.We solved the MIPs associated with `MinLin` by setting a time limit of 60 seconds. We then calculated the reduction in the size of the RMLs identified by `Greedy` and `MinLin` in comparison with the RMLs produced by `Seq`. The results are presented in Figure 6. For each plot, the x-axis shows the proportion of instances achieving a relative percentual reduction in the size of the RML that is at least as large as the value indicated in the y-axis. The figure shows that`MinLin` and `Greedy` identify smaller RMLs than `Seq` for the `mult3` and `mult4` instances. For the `vision` instances, `Seq` delivers minimum linearizations (so `MinLin` yields no gains), and `Greedy` uses approximately 15% more variables on average (see Example 4). Every sequential linearization is optimal for the `autocorr` instances, and, as a result, all algorithms provide RMLs of the same size. In terms of relative performance, the linearizations identified by `MinLin` are always at least as small as the ones produced by `Greedy`; moreover, `MinLin` is strictly better in more than 80% of the cases.



(a) 165 mult3 instances. (b) 165 mult4 instances.

Figure 6 **Reduction in the linearization size in comparison with the linearization strategy Seq.**

5.3. LP bounds

Next, we compare the LP bounds given by the RMLs constructed with the linearization strategies Seq, Greedy, MinLin, and BB. We use as baseline All, which is the RML containing McCormick inequalities for all possible triples and delivering the strongest LP bound among all RMLs. For this comparison, we calculate the root-node gap of a given linearization algorithm Alg as

$$\text{Root-node Gap}_{\texttt{Alg}} = \left(\frac{f_{\texttt{All}} - f_{\texttt{Alg}}}{\max(|f_{\texttt{All}}|, 10^{-3})} \right) \times 100 \tag{25}$$

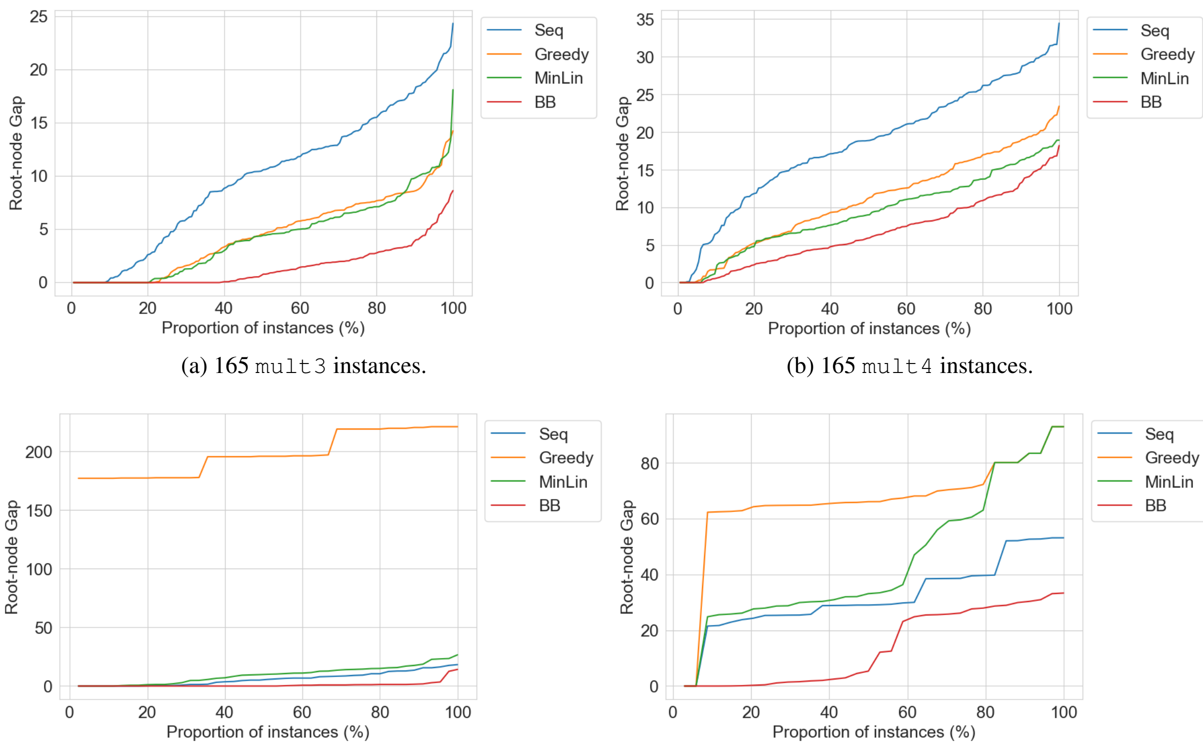
where $f_{\texttt{All}}$ and $f_{\texttt{Alg}}$ denote the bounds of the LP relaxations constructed by algorithms All and Alg, respectively. For MinLin, we use the best linearization identified within 30 seconds, providing the linearization of Greedy as warm start. For BB, we also run the algorithm for 30 seconds, using the size of the RML T identified by MinLin as parameter and T as warm start. For some instances, the MIPs built by BB are computationally challenging; in particular, if BB can only identify a suboptimal RML within 30 seconds, its LP bound may be weaker than the LP bounds of the RMLs identified by other algorithms.

The results are presented in Figure 7. For each plot, the x-axis indicates the proportion of instances for which the relative gap is less than or equal to the value indicated by the y-axis. The relative performances of the algorithms on the mult3 and mult4 instances are similar; Seq is the worst, BB is the best, and Greedy is slightly superior to MinLin. Greedy consistently delivers poor relaxation bounds for the vision instances. Figure 7d shows that small RMLs may not deliver strong bounds; in these cases, Greedy and MinLin are outperformed by Seq. Finally, BB consistently delivers tighter relaxation bounds than the other linearizations.

5.4. Comparison of MinLin and EV−MinLin

For every instance in the test set, we compare the performance of the MIP formulations used by the linearization strategies MinLin and EV−MinLin. To this end, we solved the MIPs constructed by these algorithms with a time limit of 60 seconds. The results are shown in Figure 8. This figure does not contain any performance profiles for the mult3 and vision instances because both algorithms solved the MIPs corresponding to these instances within 0.1 seconds. Moreover, for mult4, we eliminated 111 instances which can be solved trivially by both algorithms.

As seen in Figure 8, the differences in performance between MinLin and EV−MinLin are minor for the autocorr instances. More significant differences are observed for the mult4



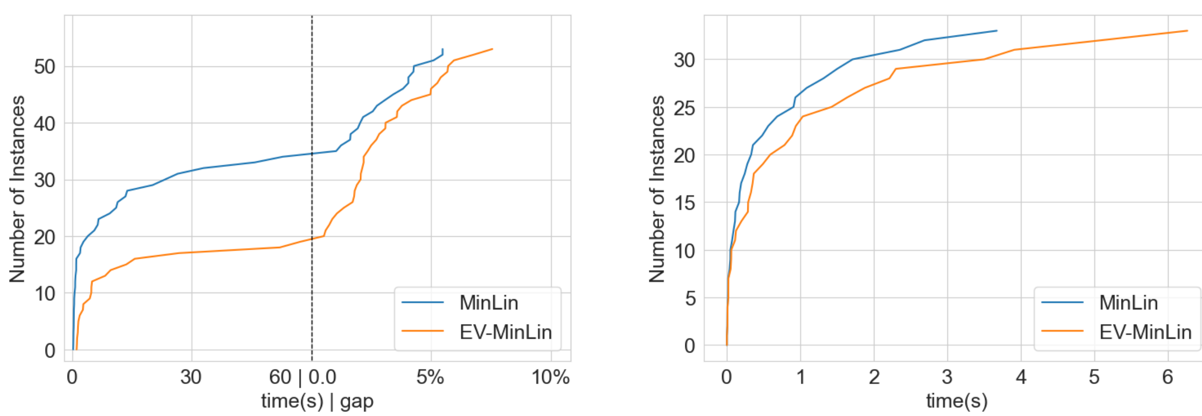
(a) 165 `mult3` instances.

(b) 165 `mult4` instances.

(c) 45 `vision` instances.

(d) 33 `autocorr` instances.

Figure 7 Root-node relaxation gaps for different linearization strategies.



(a) 54 nontrivial `mult4` instances.

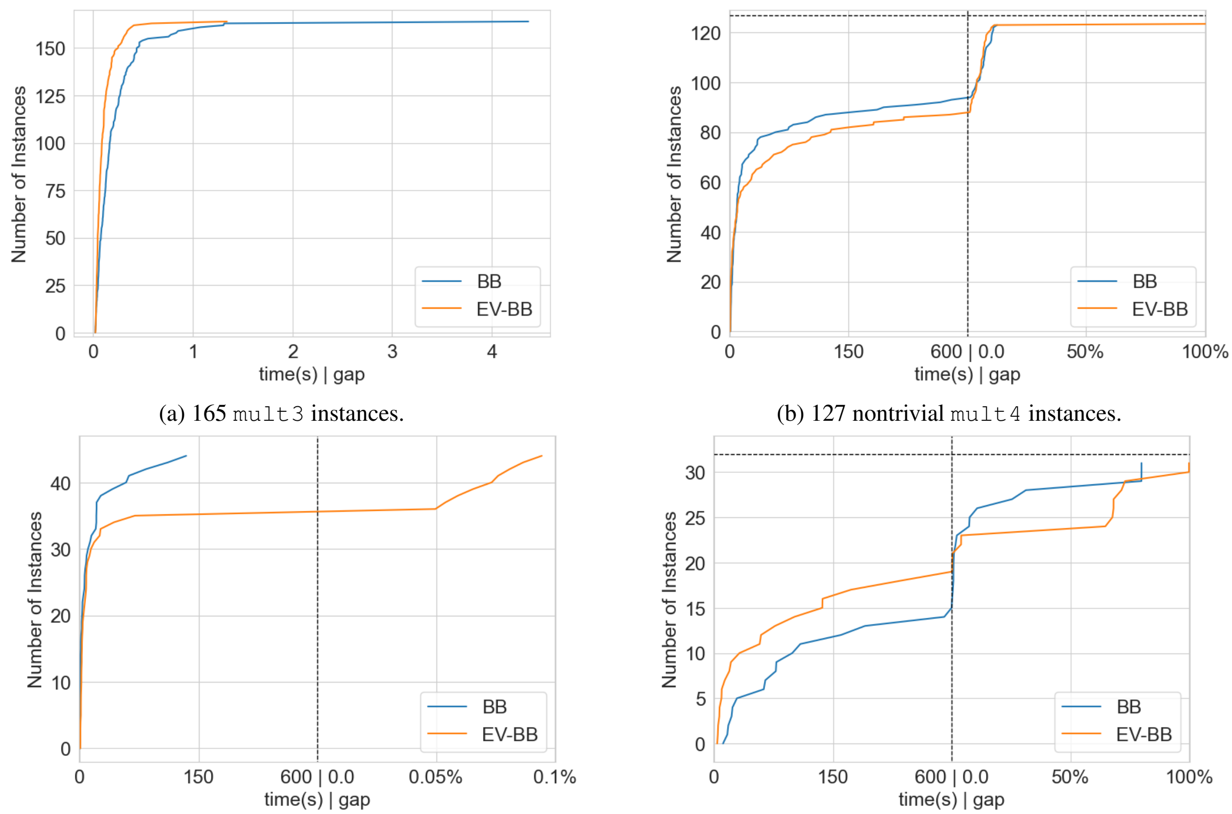
(b) 33 `autocorr` instances.

Figure 8 Performance profiles comparing the minimum-linearization MIPs built by `MinLin` and `EV-MinLin`.

instances, which are more challenging and, in some cases, cannot be solved to optimality within the time limit. For these instances, `MinLin` has a clear advantage, with more instances solved to optimality and better optimality gaps. In particular, `EV-MinLin` fails to obtain non-trivial bounds for 24 instances, whereas `MinLin` obtains gaps inferior to 10% for all instances.

5.5. Comparison of `BB` and `EV-BB`

Now, we compare the performance of the MIP formulations used by the linearization strategies `EV-BB` and `BB`. To this end, we set the size of the RMLs identified in the previous experiments as the upper bound for the number of linearization variables used by `EV-BB` and `BB`. Since the original implementation of EV−BB does not consider this upper bound, we modified the code accordingly to make EV−BB and BB comparable. We solve the MIPs with a time limit of 10 minutes. Figure 9 presents the results of these experiments. We exclude the results of 10 mult4 instances for which MinLin and EV−MinLin obtain RMLs of different sizes. We also exclude 28 mult4 instances for which the best-bound linearization MIPs built by BB and EV−BB are solved within one second.



(a) 165 mult3 instances. (b) 127 nontrivial mult4 instances. (c) 45 vision instances. (d) 32 autocorr instances.

Figure 9 Performance profiles comparing the best-bound linearization MIPs built by BB and EV−BB.

As Figure 9 indicates, the relative performance of BB and EV−BB depends on the instance. Both algorithms perform similarly for the mult3 instances. For the autocorr instances, more of the EV−BB MIPs are solved to optimality, whereas BB is better on mult4 and vision instances. Moreover, for the harder *autocorr* instances, the MIPs constructed by BB lead to smaller gaps.

5.6. Solution of the QCP and MLP formulations with global optimization solvers

In this section, we reformulate the MLPs of the test set as QCPs, by using the linearization strategies Seq, Greedy, MinLin, and BB. We then solve the resulting QCPs with GUROBI. We respectively denote these solution approaches as GUROBI−QCP−Seq, GUROBI−QCP−Greedy, GUROBI−QCP−MinLin and GUROBI−QCP−BB. We compare these solution approaches with another one which involves solving the MLPs directly with BARON; this approach is denoted by BARON−MLP. We decided to use these two solvers because GUROBI is regarded as a state-of-the-art global solver for nonconvex QCPs, whereas BARON is considered to be one of the most advanced global solvers for MLPs. As indicated by Achterberg and Towle (2020), the relaxations that GUROBI builds for nonconvex QCPs are enhanced by adding RLT cuts (Sherali and

Alameddine (1992)), SDP cuts (Sherali and Fraticelli (2002)), and facets from the Boolean Quadric Polytope (Padberg (1989)). On the other hand, the polyhedral relaxations that BARON constructs for MLPs are significantly tightened with the addition of multilinear cuts and running intersection inequalities (Del Pia et al. (2020)). The multilinear cuts correspond to facets of the convex and concave envelopes of the multilinear functions contained in the MLP. These facets can be obtained by solving an LP separation problem, whose size grows exponentially with the number of variables in the multilinear function. To manage the size of this separation problem, BARON uses a clever decomposition approach, which divides a multilinear function into lower-dimensional multilinear functions. This is coupled with a customized simplex algorithm, which efficiently solves the separation problem (Bao et al. (2015)). The running intersection inequalities are valid inequalities for the multilinear polytope and, under certain conditions, are facet-defining (Del Pia and Khajavirad (2021)). BARON generates these inequalities using a specialized polynomial-time separation algorithm (Del Pia et al. (2020)).

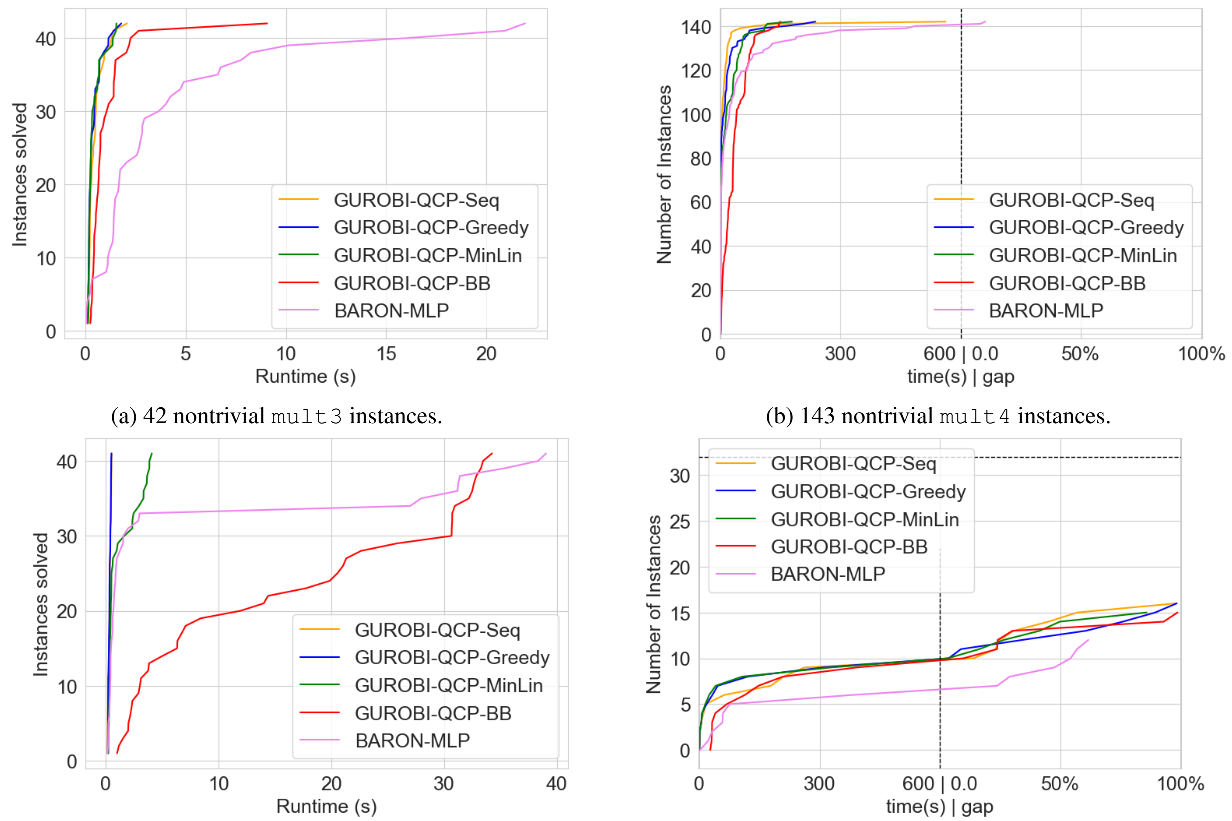
In our experiments with GUROBI and BARON, we use a time limit of 10 minutes. `GUROBI-QCP-Seq`, `GUROBI-QCP-Greedy`, and `BARON-MLP` do not need pre-processing steps. For `GUROBI-QCP-MinLin`, we retrieve the smallest RML T identified by `MinLin` within 30 seconds, using the `Greedy` RML as warm start. For `GUROBI-QCP-BB`, we obtain another RML T' by executing `BB` for 30 seconds, using T as warm start and $|T|$ as an upper bound on the number of linearization variables used by `BB`. For `GUROBI-QCP-BB`, we set the time limit to 10 minutes minus the time spent by `MinLin` and `BB`. For `GUROBI-QCP-MinLin`, we only deduct the time spent for the identification of T. Figure 10 shows the results of these experiments. For this comparison, we excluded 123 `mult3` instances, 22 `mult4` instances, 4 `vision` instances, and 1 `autocorr` instance solved by all algorithms within one second.

As seen in Figure 10, the `mult3` and `vision` instances can be considered easy since all algorithms solve them to optimality within one minute. We observe that, for some instances, the identification of high-quality RMLs becomes a bottleneck; this is particularly apparent when we observe the performance of `GUROBI-QCP-BB` on the `vision` instances. Therefore, in these cases, QCPs derived from simpler linearization strategies (i.e., those without a significant pre-processing overhead) produce better results.

The results for the `mult4` instances show that the overhead of computing linearizations is worth it for harder instances. `GUROBI-QCP-MinLin` and `GUROBI-QCP-BB` solve all `mult4` instances within 3 minutes, whereas `GUROBI-QCP-Seq` and `GUROBI-QCP-Greedy` need much more time to close the gap for all instances. `BARON-MLP` times out for some of the `mult4` instances. The `autocorr` instances are very challenging, so essentially all QCP-based algorithms have similar performance. `BARON-MLP` performs well on some of the easy `autocorr` instances, but it struggles to close the gap for harder ones. The QCP-based approaches close the gap for more `autocorr` instances than `BARON-MLP`.

Next, Figure 11 presents scatter plots comparing `GUROBI-QCP-MinLin` and `GUROBI-QCP-BB` with `BARON-MLP`. We created this figure by considering all 408 instances from test set, and by filtering out trivial instances which can be solved by all algorithms within 1 second. After filtering out such trivial instances, we obtained a collection consisting of 258 instances.

As Figure 11 indicates, `BARON-MLP` outperforms `GUROBI-QCP-MinLin` and `GUROBI-QCP-BB` for many easy instances. This shows that the overhead of computing linearizations is significant in these cases. By contrast, `GUROBI-QCP-MinLin` and `GUROBI-QCP-BB` are clearly faster than `BARON-MLP` for hard instances. In particular, for some of the hard instances for which `BARON-MLP` times out, `GUROBI-QCP-MinLin` and `GUROBI-QCP-BB` can be at least 6 times faster.



(a) 42 nontrivial `mult3` instances.

(b) 143 nontrivial `mult4` instances.

(c) 41 nontrivial `vision` instances.

(d) 32 nontrivial `autocorr` instances.

Figure 10 **Comparison between global optimization solvers. The QCPs solved by GUROBI are obtained by using different RMLs to reformulate the MLPs . BARON is used to directly solve the MLPs.**

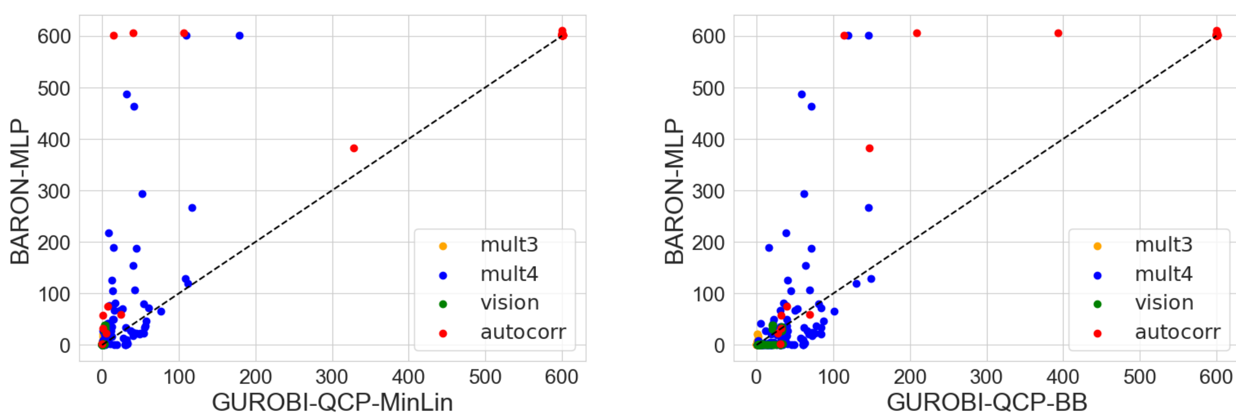


Figure 11 **Comparison between `GUROBI-QCP-MinLin` (left) and `GUROBI-QCP-BB` (right) with `BARON-MLP` for 258 nontrivial instances.**

6. Conclusions

This work presents a systematic investigation of linearization techniques for multilinear programs based on Recursive McCormick Linearizations. We design algorithms to identify optimal linearizations using two criteria: the total number of introduced variables and the strength of the LP relaxation bound. We show that the identification of a minimum-size linearization is NP-hard, and that a greedy approach to the problem can deliver arbitrarily bad results. As a result, we present an exact algorithm. We also explore structural properties of the problem to derive a MIP formulation that identifies a linearization of bounded cardinality delivering the best relaxation bound.

We compared our minimum-size and best-bound linearization strategies, `MinLin` and `BB`, with those introduced by Elloumi and Verchère (2023) (`EV-MinLin` and `EV-BB`). In our experiments, we see that `MinLin` is slightly better than `EV-MinLin`. Similarly, we observe that, for hard instances, `BB` outperforms `EV-BB`. This is explained by the fact that `BB` decomposes linearizations by bilinear terms, whereas `EV-BB` uses integrated representations of linearizations. Consequently, `EV-BB` is slightly better on easier instances, whereas `BB` obtains better bounds for harder instances. Finally, our representation of RMLs make `MinLin` and `BB` more scalable than `EV-MinLin` and `EV-BB`, respectively, in the degree of the MLPs.

Our experiments also show that, using our linearization strategies to reformulate difficult MLP instances as QCPs, and solving the resulting QCPs with GUROBI can be a much better option than directly solving the original MLPs with BARON. In these cases, our linearization strategies allow us to obtain QCP reformulations which can be significantly smaller than the QCP reformulations derived from heuristic or greedy linearization approaches. In this process, we take advantage of all the improvements that have been made in recent years to GUROBI's QCP solver. By contrast, for many easy MLP instances, directly solving the original problem with BARON leads to the best results.

References

Tobias Achterberg and Eli Towle. Non-Convex Quadratic Optimization Webinar, 2020. URL `https://www.gurobi.com/events/non-convex-quadratic-optimization/`.

M. Anthony, E. Boros, Y. Crama, and A. Gruber. Quadratic reformulations of nonlinear binary optimization problems. *Mathematical Programming*, 162:115–144, 2017.

Martin Anthony, Endre Boros, Yves Crama, and Aritanan Gruber. Quadratization of symmetric pseudo-boolean functions. *Discrete Applied Mathematics*, 203:1–12, 2016.

Xiaowei Bao, Nikolaos V. Sahinidis, and Mohit Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. *Optimization Methods and Software*, 24(4-5):485–504, 2009.

Xiaowei Bao, Aida Khajavirad, Nikolaos V Sahinidis, and Mohit Tawarmalani. Global optimization of nonconvex problems with multilinear intermediates. *Mathematical Programming Computation*, 7(1):1–37, 2015.

Jakob Bernasconi. Low autocorrelation binary sequences: Statistical mechanics and configuration space analysis. *Journal de Physique*, 48(4):559–567, 1987.

Endre Boros and Peter L Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

Endre Boros, Peter L Hammer, Michel Minoux, and David J Rader Jr. Optimal cell flipping to minimize channel density in VLSI design and pseudo-boolean optimization. *Discrete Applied Mathematics*, 90(1-3):69–88, 1999.

Endre Boros, Yves Crama, and Elisabeth Rodríguez-Heck. Compact quadratizations for pseudo-boolean functions. *Journal of combinatorial optimization*, 39:687–707, 2020.

Christoph Buchheim and Giovanni Rinaldi. Efficient reduction of polynomial zero-one optimization to the quadratic case. *SIAM Journal on Optimization*, 18(4):1398–1413, 2008.

Jonathan F Buss and Judy Goldsmith. Nondeterminism within `P*`. *SIAM Journal on Computing*, 22(3):560–572, 1993.

Carlos Cardonha, Arvind U. Raghunathan, David Bergman, and Carlos J. Nohra. Recursive McCormick linearization of multilinear programs. 2024. doi: 10.1287/ijoc.2023.0390.cd. URL `https://github.com/INFORMSJoC/2023.0390`. Available for download at https://github.com/INFORMSJoC/2023.0390.

IBM ILOG Cplex. V22.1.1: User's Manual for CPLEX. *International Business Machines Corporation*, 2022.

Y. Crama and E. Rodríguez-Heck. A class of valid inequalities for multilinear 0–1 optimization problems. *Discrete Optimization*, 25:28–47, 2017.

Yves Crama, Sourour Elloumi, Amélie Lambert, and Elisabeth Rodriguez-Heck. Quadratization and convexification in polynomial binary optimization. *HAL Open Science*, 2022. URL `https://hal.science/hal-03795395`.

Evrim Dalkiran and Laleh Ghalami. On linear programming relaxations for solving polynomial programming problems. *Computers & Operations Research*, 99:67–77, 2018.

Evrim Dalkiran and Hanif D Sherali. RLT-POS: Reformulation-Linearization Technique-based optimization software for solving polynomial programming problems. *Mathematical Programming Computation*, 8(3):337–375, 2016.

A. Del Pia and A. Khajavirad. The running intersection relaxation of the multilinear polytope. *Mathematics of Operations Research*, 46((3):1008–1037, 2021.

Alberto Del Pia, Aida Khajavirad, and Nikolaos V Sahinidis. On the impact of running intersection inequalities for globally solving polynomial optimization problems. *Mathematical programming computation*, 12(2):165–191, 2020.

Sourour Elloumi and Zoé Verchère. Efficient linear reformulations for binary polynomial optimization problems. *Computers & Operations Research*, 155:106240, 2023.

Sourour Elloumi, Amélie Lambert, and Arnaud Lazare. Solving unconstrained 0-1 polynomial programs through quadratic convex reformulation. *Journal of Global Optimization*, 80(2):231–248, 2021.

Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

Fred Glover and Eugene Woolsey. Converting the 0-1 Polynomial Programming Problem to a 0-1 Linear Program. *Operations Research*, 22(1):180–182, 1974.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022. URL `https://www.gurobi.com`.

William E Hart, Carl D Laird, Jean-Paul Watson, David L Woodruff, Gabriel A Hackebeil, Bethany L Nicholson, John D Siirola, et al. *Pyomo-optimization modeling in python*, volume 67. Springer, 2017.

KRAR Jakob and Peter Mark Pruzan. The simple plant location problem: Survey and synthesis. *European Journal of Operational Research*, 12(36-81):41, 1983.

Tanuj Karia, Claire S Adjiman, and Benoît Chachuat. Assessment of a two-step approach for global optimization of mixed-integer polynomial programs using quadratic reformulation. *Computers & Chemical Engineering*, 165: 107909, 2022.

Aida Khajavirad. On the strength of recursive McCormick relaxations for binary polynomial optimization. *Operations Research Letters*, 51:146–152, 2023.

Aida Khajavirad and Nikolaos V Sahinidis. A hybrid LP/NLP paradigm for global optimization relaxations. *Mathematical Programming Computation*, 10(3):383–421, 2018.

Mustafa R Kılınç and Nikolaos V Sahinidis. Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON. *Optimization Methods and Software*, 33(3):540–562, 2018.

JONGEUN Kim, JEAN-PHILIPPE P Richard, and MOHIT Tawarmalani. Piecewise polyhedral relaxations of multilinear optimization. *Optimization Online*, 2022.

Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

Garth P. McCormick. Computability of Global Solutions to Factorable Nonconvex Programs: Part I – Convex Underestimating Problems. *Mathematical Programming*, 10(1):147–175, dec 1976.

Ruth Misener, James B Smadbeck, and Christodoulos A Floudas. Dynamically generated cutting planes for mixed-integer quadratically constrained quadratic programs and their incorporation into GloMIQO 2. *Optimization Methods and Software*, 30(1):215–249, 2015.

Manfred Padberg. The boolean quadric polytope: Some characteristics, facets and relatives. *Mathematical programming*, 45:139–172, 1989.

Elisabeth Rodríguez-Heck. *Linear and quadratic reformulations of nonlinear optimization problems in binary variables*. Doctoral dissertation, Université de Liège, 2018.

H. S. Ryoo and N. V. Sahinidis. Analysis of bounds for multilinear functions. *J. Glob. Optim.*, 19:403–424, 2001.

Hanif D Sherali and Amine Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global optimization*, 2:379–410, 1992.

Hanif D Sherali and Barbara MP Fraticelli. Enhancing RLT relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, 22:233–261, 2002.

Hanif D Sherali and Hongjie Wang. Global optimization of nonconvex factorable programming problems. *Mathematical Programming*, 89:459–478, 2001.

Kaarthik Sundar, Harsha Nagarajan, Jeff Linderoth, Site Wang, and Russell Bent. Piecewise polyhedral formulations for a multilinear term. *Operations Research Letters*, 49(1):144–149, 2021.

Mohit Tawarmalani and Nikolaos V Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical programming*, 99(3):563–591, 2004.

Amit Verma and Mark Lewis. Optimal quadratic reformulations of fourth degree pseudo-boolean functions. *Optimization Letters*, 14:1557–1569, 2020.

Keith Zorn and Nikolaos V Sahinidis. Global optimization of general nonconvex problems with intermediate polynomial substructures. *Journal of Global Optimization*, 59:673–693, 2014.