A class of valid inequalities for multilinear 0-1 optimization problems

Yves Crama*[1] and Elisabeth Rodríguez-Heck†[1]

[1]QuantOM, HEC Management School, University of Liège, Belgium

May 9, 2016

Abstract

This paper investigates the polytope associated with the classical standard linearization technique for the unconstrained optimization of multilinear polynomials in 0-1 variables. A new class of valid inequalities, called 2-links, is introduced to strengthen the LP relaxation of the standard linearization. The addition of the 2-links to the standard linearization inequalities provides a complete description of the convex hull of integer solutions for the case of functions consisting of at most two nonlinear monomials. For the general case, various computational experiments show that the 2-links improve both the standard linearization bound and the computational performance of exact branch & cut methods. The improvements are especially significant for a class of instances inspired from the image restoration problem in computer vision. The magnitude of this effect is rather surprising in that the 2-links are in relatively small number (quadratic in the number of terms of the objective function).

Keywords— multilinear binary optimization, pseudo-Boolean optimization, integer nonlinear programming, standard linearization

*yves.crama@ulg.ac.be

†elisabeth.rodriguezheck@ulg.ac.be

1 Introduction

We consider the problem of optimizing multilinear polynomials defined on binary variables, with no additional constraints. More precisely, consider n binary variables x_i, $i \in [n] = \{1, \ldots, n\}$. Let $2^{[n]}$ be the set of subsets of indices in $[n]$, and denote by a_S a real value associated with every $S \in [n]$. When $|S| = 1$, we write a_i instead of $a_{\{i\}}$ for simplicity. Let $\mathcal{S} \subseteq 2^{[n]}$ be the set of subsets S such that $a_S \neq 0$ and $|S| \geq 2$. Then, the multilinear polynomial expression

$$f(x_1, \ldots, x_n) = \sum_{S \in \mathcal{S}} a_S \prod_{i \in S} x_i + \sum_{i \in [n]} a_i x_i \tag{1}$$

defines a *pseudo-Boolean function*, that is, a mapping $f : \{0, 1\}^n \to \mathbb{R}$ that assigns a real value to each tuple of n binary variables $(x_1, \ldots, x_n)$. Conversely, it is known that every pseudo-Boolean function f can be represented uniquely by a multilinear polynomial of the form (1) (see [22, 23, 11]).

We are interested in optimizing functions of the form (1) over $\{0, 1\}^n$. This problem is known to be $\mathcal{NP}$-hard, even when the objective function is quadratic (in which case it equivalent with max-cut; see [13]). More generally, multilinear binary optimization belongs to the field of pseudo-Boolean optimization, which has been extensively studied during the last century and especially in the last 50 years, given its applicability to a wide range of areas such as reliability theory, computer science, statistics, economics, finance, operations research, management science, discrete mathematics, or computer vision (see [5] and [11] for a list of applications and references).

Several approaches have been proposed to solve the multilinear binary optimization problem, such as reductions to the linear or to the quadratic case, algebraic methods, enumerative methods like branch-and-bound and its variants, or cutting-plane methods (see, for example, surveys [5, 8, 12, 24, 25]). The efficacy of these techniques strongly depends on the structure of the problem, and it is unclear whether one approach is generally better than the others. In this paper we focus on linearization, an approach that attempts to draw benefit from the extensive literature on integer linear programming. However, linearization techniques present two important drawbacks: they introduce many additional variables and constraints, and the resulting continuous relaxation usually leads to weak bounds. Our objective is to palliate the latter drawback by introducing new inequalities that tighten the formulation of the linearized problem.

The *standard linearization* is a classical linearization procedure which consists in substituting each nonlinear monomial $\prod_{i \in S} x_i$ by a new variable y_S, and imposing $y_S = \prod_{i \in S} x_i$ as a constraint for all $S \in \mathcal{S}$. We denote by X_{SL} the set of binary points satisfying these constraints, that is,

$$X_{SL} = \{(x, y) \in \{0, 1\}^{n+|\mathcal{S}|} \mid y_S = \prod_{i \in S} x_i, \ \forall S \in \mathcal{S}\}, \tag{2}$$

and we denote its convex hull by P^*_{SL}:

$$P^*_{SL} = conv(X_{SL}). \tag{3}$$

Then, the problem of optimizing (1) is equivalent to the linear programming problem

$$\min_{(x,y) \in P^*_{SL}} L_f(x, y) = \sum_{S \in \mathcal{S}} a_S y_S + \sum_{i \in [n]} a_i x_i. \tag{4}$$

In order to obtain a 0–1 linear programming formulation of our problem, the polynomial equation $y_S = \prod_{i \in S} x_i$ can be expressed using the following constraints, to be called *standard linearization inequalities* in the sequel:

$$y_S \le x_i, \qquad \forall i \in S \tag{5}$$

$$y_S \ge \sum_{i \in S} x_i - (|S| - 1), \tag{6}$$

$$y_S \ge 0 \tag{7}$$

More precisely, when x_i is binary for all $i \in S$, the feasible solutions of the constraints (5)–(7) are exactly the solutions of the polynomial equation $y_S = \prod_{i \in S} x_i$. (The integrality requirement does not need to be explicitly stated for y_S: when the original variables x_i are binary, then y_S automatically takes a binary value too.) So, if we define the *standard linearization polytope* associated with f as

$$P_{SL} = \{(x, y) \in [0, 1]^{n+|\mathcal{S}|} \mid (5), (6), \ \forall S \in \mathcal{S}\}, \tag{8}$$

then P_{SL} is a valid formulation of X_{SL} in the sense that X_{SL} is exactly the set of binary points in P_{SL}.

The standard linearization was proposed by several authors independently ([18, 19, 32, 33]), in a slightly different form from (5)–(7) and with integrality constraints on the variables y_S. The initial formulation was later improved by Glover and Woolsey, in a first contribution by adding fewer constraints and variables in the reformulation [20], and in a second contribution by introducing continuous auxiliary variables rather than integer ones [21].

When f contains a single nonlinear monomial, P_{SL} is equal to P^*_{SL} (see Section 2). However, for the general case when f contains an arbitrary number of nonlinear monomials, finding a concise perfect formulation of P^*_{SL} is probably hopeless (unless $\mathcal{P} = \mathcal{NP}$). Recent work concerning polyhedral descriptions of the standard linearization polytope can be found in [7, 15, 14].

In this paper, we introduce a new class of valid inequalities for P^*_{SL}, that we call *2-links*.

Our main contribution is that, when f contains exactly two nonlinear monomials, a complete formulation of P^*_{SL} is obtained by adding the 2-link inequalities to the standard linearization constraints of P_{SL}. We also establish that the 2-links are facet-defining when f consists of nested monomials, that is, of a chain of monomials contained in each other. Furthermore, we provide computational experiments showing that for various classes of multilinear polynomials, adding the 2-links to the standard formulation P_{SL} provides significant improvements in the quality of the bounds of the linear relaxations and on the performance of exact resolution methods.

The rest of the paper is structured as follows. Section 2 formally introduces the 2-links. Section 3 establishes their strength for the case of nested monomials, and derives some related properties of the standard linearization inequalities. Section 4 presents our main result for the case of two nonlinear monomials. Section 5 describes our computational experiments. Finally, Section 6 proposes some conclusions and sketches further research questions.

2 Definition and validity of 2-link inequalities

This section formally introduces the 2-links and establishes some of their properties. Let f be the function on variables x_i, $i \in [n]$, represented by the multilinear polynomial (1) with $a_S \neq 0$ for all $S \in \mathcal{S}$. Let the set X_{SL}, its convex hull P^*_{SL}, and its standard linearization polytope P_{SL} be defined as in Section 1. Note that X_{SL}, P_{SL} and P^*_{SL} actually depend on f, or more precisely on the set of monomials $\mathcal{S}$. However we do not indicate this dependence in the notation for simplicity.

As in [9], we say that a polytope P (and by extension, any system of linear inequalities defining P) is a *perfect formulation* of a set X if P is exactly the convex hull of X.

Remark 1. *When $\mathcal{S}$ contains a single nonlinear monomial S, the inequalities (5)–(7) and the bound constraints $0 \leq x_i \leq 1$ ($i \in [n]$) provide a perfect formulation of X_{SL}. That is, $P^*_{SL} = P_{SL}$.*

This remark appears to be part of the folklore of the field of nonlinear binary optimization. It can be easily derived by direct arguments, and it also follows from related results, e.g., by McCormick [30] and by Al-Khayyal and Falk [2] for the quadratic case, by Crama [10] and by Ryoo and Sahinidis [31] for the general case of degree higher than two (see also [29]).

However, P_{SL} provides a very weak relaxation of P^*_{SL} when f contains an arbitrary number of nonlinear monomials. We now provide inequalities that tighten this continuous relaxation.

Definition 1. *Consider two monomials indexed by subsets $S, T \in \mathcal{S}$ and consider variables y_S, y_T such that $y_S = \prod_{i\in S} x_i$, $y_T = \prod_{i\in T} x_i$. The* 2-link *associated with (S,T) is the linear inequality*

$$y_S \leq y_T - \sum_{i\in T\setminus S} x_i + |T\setminus S|. \tag{9}$$

Proposition 1. Validity of the 2-links. *For any $S, T \in \mathcal{S}$, the 2-link inequality (9) is valid for P^*_{SL}.*

Proof. It suffices to show that (9) is satisfied by all points (x, y) in X_{SL}. This is trivial when $y_S \leq y_T$. When $(y_S, y_T) = (1, 0)$, the monomial $\prod_{i\in S} x_i$ takes value one and the monomial $\prod_{i\in T} x_i$ takes value zero, which implies that a variable in $T\setminus S$ must be zero. It follows again that (9) is satisfied. □

Note that the 2-link inequalities are valid when $|S \cap T| < 2$, but in that case they do not strengthen the relaxation of P_{SL}. Indeed, when $S \cap T = \emptyset$, then (9) can be derived by simply adding the standard linearization inequalities $y_S \leq 1$ and $\sum_{i\in T} x_i - (|T| - 1) \leq y_T$. Also, when $|S \cap T| = 1$, say, $S \cap T = \{k\}$, then (9) is obtained by adding up $y_S \leq x_k$ and $\sum_{i\in T} x_i - (|T| - 1) \leq y_T$.

3 Nested nonlinear monomials

In order to illustrate the strength of 2-link inequalities, we next establish a result (Proposition 2 hereunder) concerning multilinear functions with a particular structure, namely, those for which the nonlinear monomials are nested. We already note that Proposition 2 has been independently found by Fischer, Fischer and McCormick [16] in the more general framework of polynomial functions optimized over a matroid polytope; we will return to this remark at the end of the section.

Observe that the 2-link inequality associated with (S,T) takes the form $y_S \leq y_T$ when $T \subseteq S$.

Proposition 2. Nested monomials. *Consider a function*

$$f(x) = \sum_{k\in[l]} a_{S^{(k)}} \prod_{i\in S^{(k)}} x_i + \sum_{i\in S^{(l)}} a_i x_i$$

defined on l monomials such that $S^{(1)} \subset S^{(2)} \subset \cdots \subset S^{(l)}$, where $|S^{(1)}| \geq 2$ and $S^{(l)} = [n]$ without loss of generality. Let $P^{*,nest}_{SL}$ be the convex hull of the integer *points of the standard linearization polytope associated with f. Then, the 2-links*

$$y_{S^{(k)}} \le y_{S^{(k+1)}} - \sum_{i \in S^{(k+1)} \setminus S^{(k)}} x_i + |S^{(k+1)} \setminus S^{(k)}|, \tag{10}$$

$$y_{S^{(k+1)}} \le y_{S^{(k)}}, \tag{11}$$

for $k = 1, \ldots, l-1$, are facet-defining for $P_{SL}^{*,nest}$.

Proof. Let $x = (x_1, \ldots, x_n)$, let u_i be the n-dimensional unit vector with i^{th} component equal to one, let $y = (y_{S^{(1)}}, \ldots, y_{S^{(l)}})$, and let v_j be the l-dimensional unit vector with the j^{th} component equal to one.

Observe first that $P_{SL}^{*,nest}$ is full-dimensional; indeed, the n points $(x, y) = (u_i, 0)$, $\forall i \in [n]$, the l points $(x, y) = (\sum_{i \in S^{(k)}} u_i, \sum_{j \le k} v_j)$, $\forall k \in [l]$, and the point $(0, 0)$ are in $P_{SL}^{*,nest}$ and are affinely independent.

Let F be the face of $P_{SL}^{*,nest}$ represented by (10), $F = \{(x, y) \in P_{SL}^{*,nest} \mid y_{S^{(k)}} = y_{S^{(k+1)}} - \sum_{i \in S^{(k+1)} \setminus S^{(k)}} x_i + |S^{(k+1)} \setminus S^{(k)}|\}$, for a fixed $k < l$. To prove that F is a facet, we will show that F is contained in a unique hyperplane and thus $dim(F) = dim(P_{SL}^{*,nest}) - 1$, since $P_{SL}^{*,nest}$ is full-dimensional. Consider $b(x, y) = \sum_{i \in [n]} b_i x_i + \sum_{k \in [l]} b_{S^{(k)}} y_{S^{(k)}}$ and assume that F is contained in the hyperplane $b(x, y) = b_0$. We will see that this is only possible if $b(x, y) = b_0$ is a multiple of

$$y_{S^{(k)}} = y_{S^{(k+1)}} - \sum_{i \in S^{(k+1)} \setminus S^{(k)}} x_i + |S^{(k+1)} \setminus S^{(k)}|. \tag{12}$$

1. The point $(x, y) = (\sum_{i \in S^{(k+1)} \setminus S^{(k)}} u_i, 0)$ is in F. Assuming that (x, y) satisfies $b(x, y) = b_0$, we have that $\sum_{i \in S^{(k+1)} \setminus S^{(k)}} b_i = b_0$.

2. Fix an index $j \in S^{(k)}$, and consider $(x, y) = (u_j + \sum_{i \in S^{(k+1)} \setminus S^{(k)}} u_i, 0) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$, we have $b_j + \sum_{i \in S^{(k+1)} \setminus S^{(k)}} b_i = b_0$, which implies, together with the previous condition, that $b_j = 0$, $\forall j \in S^{(k)}$.

3. If $k+1 < l$, fix an index $j \in S^{(l)} \setminus S^{(k+1)}$, and consider $(x, y) = (\sum_{i \in S^{(k+1)} \setminus S^{(k)}} u_i + u_j, 0) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$, we have that $b_j + \sum_{i \in S^{(k+1)} \setminus S^{(k)}} b_i = b_0$, which implies together with the first condition that $b_j = 0$, $\forall j \in S^{(l)} \setminus S^{(k+1)}$.

4. We next show that $b_{S^{(j)}} = 0$ for $j < k$. Assume first that $j = 1 < k$ and let $(x, y) = (\sum_{i \in S^{(k+1)} \setminus S^{(k)}} u_i + \sum_{i \in S^{(1)}} u_i, v_1) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$, we have that $\sum_{i \in S^{(1)}} b_i + \sum_{i \in S^{(k+1)} \setminus S^{(k)}} b_i + b_{S^{(1)}} = b_0$, which implies, together with the previous conditions, that $b_{S^{(1)}} = 0$. Repeating this procedure for $j = 2, \ldots, k-1$ (in this order), we obtain that $b_{S^{(j)}} = 0$ for all $j < k$.

5. Fix a $j \in S^{(k+1)} \backslash S^{(k)}$, and take $(x, y) = (\sum_{i \in S^{(k+1)} \backslash \{j\}} u_i, \sum_{i \leq k} v_i) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$, we have $\sum_{i \in S^{(k+1)}, i \neq j} b_i + \sum_{i \leq k} b_{S^{(i)}} = b_0$, which implies, together with the previous conditions and repeating for $j \in S^{(k+1)} \backslash S^{(k)}$, that $b_j = b_{S^{(k)}}$, for all $j \in S^{(k+1)} \backslash S^{(k)}$.

6. Consider $(x, y) = (\sum_{i \in S^{(k+1)}} u_i, \sum_{i \leq k+1} v_i) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$, we obtain that $\sum_{i \in S^{(k+1)}} b_i + \sum_{i \leq k+1} b_{S^{(i)}} = b_0$, which implies, together with the previous conditions, that $b_{S^{(k)}} + b_{S^{(k+1)}} = 0$.

7. Consider subset $S^{(k+2)}$, and take $(x, y) = (\sum_{i \in S^{(k+2)}} u_i, \sum_{j \leq k+2} v_j) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$, and using the previous conditions we have that $b_{S^{(k+2)}} = 0$. Repeating this reasoning for $j = k + 3, \ldots l$ (in this order), we obtain $b_{S^{(j)}} = 0$, for all $j > k + 1$.

Putting together the previous conditions, we have that $b(x, y) = b_0$ takes the form $b_{S^{(k)}} \sum_{i \in S^{(k+1)} \backslash S^{(k)}} x_i + b_{S^{(k)}} y_{S^{(k)}} - b_{S^{(k)}} y_{S^{(k+1)}} = |S^{(k+1)} \backslash S^{(k)}| \, b_{S^{(k)}}$, which is a multiple of equation (12) as required.

In a similar way, it can be proved that the face represented by (11) is a facet. □

Remark 2. *The 2-link inequalities are only facet-defining for consecutive monomials in the nested sequence. In fact, the 2-links corresponding to non-consecutive monomials are implied by the 2-links associated with consecutive monomials.*

The following remarks can be proved using similar arguments as those presented in the proof of Proposition 2. They imply, in particular, that the standard linearization inequalities (5)–(7) are not always facet-defining for $P^{*,nest}_{SL}$.

Remark 3. *The lower bounding inequality* $0 \leq y_{S^{(l)}}$ *is facet-defining for* $P^{*,nest}_{SL}$. *However, the inequalities* $0 \leq y_{S^{(k)}}$, $k = 1, \ldots, l-1$ *are redundant, since they are implied by* $0 \leq y_{S^{(k+1)}}$ *and by* $y_{S^{(k+1)}} \leq y_{S^{(k)}}$.

Remark 4. *The standard linearization inequality* $y_{S^{(1)}} \geq \sum_{i \in S^{(1)}} x_i - (|S^{(1)}| - 1)$ *is facet-defining for* $P^{*,nest}_{SL}$. *However* $y_{S^{(k)}} \geq \sum_{i \in S^{(k)}} x_i - (|S^{(k)}| - 1)$, $k = 2, \ldots, l$ *are redundant, since they are implied by* $y_{S^{(k-1)}} \geq \sum_{i \in S^{(k-1)}} x_i - (|S^{(k-1)}| - 1)$ *and* $y_{S^{(k-1)}} \leq y_{S^{(k)}} - \sum_{i \in S^{(k)} \backslash S^{(k-1)}} x_i + |S^{(k)} \backslash S^{(k-1)}|$.

Remark 5. *The standard linearization inequalities* $y_{S^{(k)}} \leq x_i$, $i \in S^{(k)} \backslash S^{(k-1)}$ *are facet-defining for* $P^{*,nest}_{SL}$ *for all* $k = 1, \ldots, l$, *where* $S^{(0)} = \emptyset$. *However* $y_{S^{(k)}} \leq x_i$, $i \in S^{(k-1)}$ *are redundant, since they are implied by* $y_{S^{(k-1)}} \leq x_i$, $i \in S^{(k-1)}$ *and* $y_{S^{(k)}} \leq y_{S^{(k-1)}}$.

The results by Fischer et al. [16] actually imply that inequalities (10), (11), together with the facet-defining inequalities of Remarks 3, 4 and 5 , define the convex hull $P^{*,nest}_{SL}$ for the nested case.

4 The case of two nonlinear monomials

In this section, we present some results for the special case of a multilinear function $f(x) = a_S \prod_{i\in S} x_i + a_T \prod_{i\in T} x_i + \sum_{i\in[n]} a_i x_i$ containing exactly two nonlinear monomials indexed by S and T. Our first result states that the 2-links associated with S and T are facet-defining for P^*_{SL}, whenever $|S \cap T| \geq 2$. As observed in Section 2, the 2-links are valid but redundant for $|S \cap T| < 2$. Our second and main result is a theorem stating that the 2-links, together with the standard linearization inequalities, provide a complete description of P^*_{SL}. Throughout this section we assume that $S \cup T = [n]$ for simplicity. The results provided can be easily extended to the more general case $S \cup T \subseteq [n]$, since the variables in $[n]\backslash(S \cup T)$ do not complicate the description of the convex hull. Also, since the case of nested monomials has been covered in the previous section, we assume that $S \nsubseteq T$ and $T \nsubseteq S$.

Remark 6. *The standard linearization inequalities (5), (6) and (7) are facet-defining for the case of a function f containing exactly two nonlinear monomials defined by subsets S and T such that $S \nsubseteq T$ and $T \nsubseteq S$.*

This remark can be proved using similar arguments as in the proof of Proposition 2; it is valid for $|S \cap T| \geq 0$.

Proposition 3. *The 2-links*

$$y_S \leq y_T - \sum_{i\in T\backslash S} x_i + |T\backslash S| \tag{13}$$

$$y_T \leq y_S - \sum_{i\in S\backslash T} x_i + |S\backslash T|, \tag{14}$$

are facet-defining for P^*_{SL}, the convex hull of the integer points of the standard linearization polytope associated with a function f containing exactly two nonlinear monomials defined by subsets S and T such that $|S \cap T| \geq 2$.

Proof. Let $x = (x_1, \ldots, x_n)$, let u_i denote the n-dimensional unit vector with i^{th} component equal to one, let $y = (y_S, y_T)$, and let $v_S = (1, 0)$, $v_T = (0, 1)$, respectively. Since Proposition 2 covers the case of nested monomials, we assume that $S \nsubseteq T$ and $T \nsubseteq S$. We will prove that (13) is facet-defining (the proof for (14) is analogous).

Observe that P^*_{SL} is full-dimensional (i.e., of dimension $n+2$), given that the n points $(u_i, 0)$, $\forall i \in [n]$, the two points $(\sum_{i\in S} u_i, v_S)$ and $(\sum_{i\in T} u_i, v_T)$, and the point $(0, 0)$ are contained in P^*_{SL} and are affinely independent.

Now, let F be the face of P^*_{SL} represented by (13), $F = \{(x, y) \in P^*_{SL} \mid y_S = y_T - \sum_{i\in T\backslash S} x_i + |T\backslash S|\}$. Let $b(x, y) = \sum_{i\in[n]} b_i x_i + b_S y_S + b_T y_T$ and assume that F

is contained in the hyperplane $b(x, y) = b_0$. We will use the same technique as for Proposition 2 to see that F is a facet.

1. Consider $(x, y) = (\sum_{i\in T\setminus S} u_i, 0) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$, we obtain that $\sum_{i\in T\setminus S} b_i = b_0$.

2. Fix an index $j \in S$ and consider $(x, y) = (\sum_{i\in T\setminus S} u_i + u_j, 0) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$ and using the previous condition we deduce that $b_j = 0$ for all $j \in S$.

3. Fix an index $j \in T\setminus S$. Consider $(x, y) = (\sum_{i\in(S\cup T)\setminus\{j\}} u_i, v_S) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$ we obtain $\sum_{i\in(S\cup T)\setminus\{j\}} b_i + b_S = b_0$, which, together with the previous conditions, implies $b_j = b_S$ for all $j \in T\setminus S$.

4. Consider $(x, y) = (\sum_{i\in S\cup T} u_i, v_S + v_T) \in F$. Assuming that (x, y) satisfies $b(x, y) = b_0$, and together with the previous conditions, we obtain $b_S = -b_T$.

Putting together the previous conditions, we have that $b(x, y) = b_0$ takes the form $b_S y_S - b_S y_T + b_S \sum_{i\in T\setminus S} x_i = b_S|T\setminus S|$. □

Proposition 3 establishes that the 2-links are strong valid inequalities for P^*_{SL}. We will see that, in addition, when we add the 2-links to P_{SL}, we obtain a complete description of P^*_{SL}. For this, let

$$P^{2links}_{SL} = P_{SL} \cap \{(x, y_S, y_T) \in \mathbb{R}^{n+2} \mid (13),\ (14) \text{ are satisfied}\}.$$

It is easy to see that the bound constraints $y_S \leq 1$, $y_T \leq 1$ and $x_i \geq 0$, $\forall i \in [n]$ are implied by the standard linearization inequalities (5) and by the remaining bound constraints. We keep them in the description of P^{2links}_{SL} for simplicity of exposition. Note that for $|S \cap T| < 2$, we have that $P^{2links}_{SL} = P_{SL}$, since the 2-links are redundant.

Theorem 1. *$P^*_{SL} = P^{2links}_{SL}$ when the function f contains two nonlinear monomials.*

For disjoint monomials, this can be derived directly from Remark 1. For the general case, the proof relies on a classical result by Balas ([3, 4], see also [9] for the bounded case) aimed at modeling the convex hull of the union of q polytopes $P^1, \ldots, P^q \subseteq \mathbb{R}^m$ such that, for $k \in [q]$, P^k is described by the inequalities

$$\begin{aligned} A^k x &\leq b^k, \\ 0 \leq x &\leq d^k. \end{aligned} \tag{15}$$

The union $\cup_{k\in[q]} P^k$ can be modeled by introducing q binary variables z^k, indicating whether a point x is in the k^{th} polytope, and q vectors of variables $x^k \in \mathbb{R}^m$. Then, a point $x \in \mathbb{R}^m$ belongs to $\cup_{k\in[q]} P_k$ if and only if there exist $x^1, \dots, x^q$ and $z^1, \dots, z^q$ such that

$$\sum_{k\in[q]} x^k = x \tag{16}$$

$$A^k x^k \le b^k z^k, \quad k \in [q] \tag{17}$$

$$0 \le x^k \le d^k z^k, \quad k \in [q] \tag{18}$$

$$\sum_{k\in[q]} z^k = 1 \tag{19}$$

$$z^k \in \{0,1\}, \quad k \in [q]. \tag{20}$$

Let Q be the set of points $(x, x^1, \dots, x^q, z^1, \dots, z^q)$ satisfying (16)–(20). Balas' result states that this disjunctive model is perfect:

Proposition 4. *[3, 4, 9] The convex hull of solutions to (16)–(20), that is, $conv(Q)$, is described by inequalities (16)–(19) and $z^k \in [0,1]$ for $k \in [q]$.*

For any set $W \subseteq \mathbb{R}^{n+l}$ (defined on variables $(x, w) \in \mathbb{R}^{n+l}$), let $Proj_x(W)$ be the projection of W on the space of the x variables. With these notations we can write the union of the polytopes as $\cup_{k\in[q]} P^k = Proj_x(Q)$ and, by commutativity of the operators $conv$ and $Proj_x$, we have that

$$conv(\cup_{k\in[q]} P^k) = Proj_x(conv(Q)). \tag{21}$$

So, Proposition 4 provides a perfect extended formulation of $conv(\cup_{k\in[q]} P^k)$.

We are now ready for a proof of Theorem 1.

Proof. We will show that all vertices of P^{2links}_{SL} are integer and therefore P^{2links}_{SL} is a perfect formulation of P^*_{SL} (i.e., $P^{2links}_{SL} = P^*_{SL}$). Consider the following set of inequalities, where (22)–(23) result from the standard linearization of $y_{S\cap T} = \prod_{i\in S\cap T} x_i$, (24)–(26) result from the standard linearization of $y_S = y_{S\cap T} \prod_{i\in S\setminus T} x_i$, and (27)–(29) result from the standard linearization of $y_T = y_{S\cap T} \prod_{i\in T\setminus S} x_i$:

$$y_{S\cap T} \le x_i, \qquad \forall i \in S\cap T, \tag{22}$$

$$y_{S\cap T} \ge \sum_{i\in S\cap T} x_i - (|S\cap T| - 1), \tag{23}$$

$$y_S \le y_{S\cap T}, \tag{24}$$

$$y_S \le x_i, \qquad \forall i \in S\setminus T, \tag{25}$$

$$y_S \ge \sum_{i\in S\setminus T} x_i + y_{S\cap T} - |S\setminus T|, \tag{26}$$

$$y_T \le y_{S\cap T}, \tag{27}$$

$$y_T \leq x_i, \qquad \forall i \in T \setminus S, \tag{28}$$

$$y_T \geq \sum_{i \in T \setminus S} x_i + y_{S \cap T} - |T \setminus S|, \tag{29}$$

$$0 \leq y_S \leq 1, \tag{30}$$

$$0 \leq y_T \leq 1, \tag{31}$$

$$0 \leq y_{S \cap T} \leq 1, \tag{32}$$

$$0 \leq x_i \leq 1, \qquad \forall i \in S \cup T. \tag{33}$$

Let P denote the polytope

$$P = \{(x, y_S, y_T, y_{S \cap T}) \in \mathbb{R}^{n+3} \mid (22) - (33) \text{ are satisfied}\},$$

and let P^0 (respectively, P^1) denote the faces of P defined by fixing $y_{S \cap T} = 0$ (respectively, $y_{S \cap T} = 1$) in (22)–(33). So, P^0 is described by the constraints

$$\begin{aligned}
&\sum_{i \in S \cap T} x_i - (|S \cap T| - 1) \leq 0 \\
&y_{S \cap T} = y_S = y_T = 0 \\
&0 \leq x_i \leq 1, \qquad \forall i \in S \cup T
\end{aligned}$$

and P^1 is described by

$$\begin{aligned}
&\sum_{i \in S \setminus T} x_i - (|S \setminus T| - 1) \leq y_S \\
&\sum_{i \in T \setminus S} x_i - (|T \setminus S| - 1) \leq y_T \\
&y_S \leq x_i, \qquad \forall i \in S \setminus T \\
&y_T \leq x_i, \qquad \forall i \in T \setminus S \\
&0 \leq y_S \leq 1 \\
&0 \leq y_T \leq 1 \\
&y_{S \cap T} = x_i = 1, \qquad \forall i \in S \cap T \\
&0 \leq x_i \leq 1, \qquad \forall i \in (S \setminus T) \cup (T \setminus S)\}.
\end{aligned}$$

Observe that P^0 is an integer polytope because it is defined by a (totally unimodular) cardinality constraint. Polytope P^1 is also integer, because it is defined by the standard linearization constraints corresponding to two monomials on disjoint sets of variables, namely, $\prod_{i \in S \setminus T} x_i$ and $\prod_{i \in T \setminus S} x_i$; hence we can use the fact that P_{SL} is a perfect formulation for a single nonlinear monomial.

As a consequence, $conv(P^0 \cup P^1)$ also is an integral polytope. Our objective is now to describe this polytope and, namely, to show that $conv(P^0 \cup P^1) = P$. In view of Proposition 4, a point $(x, y_S, y_T, y_{S\cap T})$ belongs to $conv(P^0 \cup P^1)$ if and only if there exist $x_i^0, x_i^1 \in \mathbb{R}^n$ $(i \in S \cup T)$ and $y_S^0, y_S^1, y_T^0, y_T^1, y_{S\cap T}^0, y_{S\cap T}^1, z^0, z^1 \in \mathbb{R}$ such that

$$x_i^0 + x_i^1 = x_i, \qquad \forall i \in S \cup T, \tag{34}$$

$$y_S^0 + y_S^1 = y_S, \tag{35}$$

$$y_T^0 + y_T^1 = y_T, \tag{36}$$

$$y_{S\cap T}^0 + y_{S\cap T}^1 = y_{S\cap T}, \tag{37}$$

$$\sum_{i \in S\cap T} x_i^0 \leq (|S \cap T| - 1)\, z^0, \tag{38}$$

$$y_{S\cap T}^0 = 0, \tag{39}$$

$$y_S^0 = 0, \tag{40}$$

$$y_T^0 = 0, \tag{41}$$

$$x_i^0 \leq z^0, \qquad \forall i \in S \cup T, \tag{42}$$

$$0 \leq x_i^0, \qquad \forall i \in S \cup T, \tag{43}$$

$$\sum_{i \in S\setminus T} x_i^1 - y_S^1 \leq (|S\setminus T| - 1)\, z^1, \tag{44}$$

$$\sum_{i \in T\setminus S} x_i^1 - y_T^1 \leq (|T\setminus S| - 1)\, z^1, \tag{45}$$

$$y_S^1 \leq x_i^1, \qquad \forall i \in S\setminus T, \tag{46}$$

$$y_T^1 \leq x_i^1, \qquad \forall i \in T\setminus S, \tag{47}$$

$$y_{S\cap T}^1 = z^1, \tag{48}$$

$$y_S^1 \leq z^1, \tag{49}$$

$$0 \leq y_S^1, \tag{50}$$

$$y_T^1 \leq z^1, \tag{51}$$

$$0 \leq y_T^1, \tag{52}$$

$$x_i^1 = z^1, \qquad \forall i \in S \cap T, \tag{53}$$

$$x_i^1 \leq z^1, \qquad \forall i \in (S\setminus T) \cup (T\setminus S) \tag{54}$$

$$0 \leq x_i^1, \qquad \forall i \in (S\setminus T) \cup (T\setminus S) \tag{55}$$

$$z^0 + z^1 = 1, \tag{56}$$

$$z^0 \leq 1, \tag{57}$$

$$0 \leq z^0, \tag{58}$$

$$z^1 \leq 1, \tag{59}$$
$$0 \leq z^1. \tag{60}$$

Let W denote the polytope defined by constraints (34)-(60). We will explicitly calculate the projection $Proj_{(x,y_S,y_T,y_{S\cap T})}(W) = conv(P^0 \cup P^1)$.

First, we simplify constraints (34)-(60) using the following observations:

- Substituting (40) in (35) we obtain $y_S^1 = y_S$.
- Substituting (41) in (36) we obtain $y_T^1 = y_T$.
- Substituting (39) in (37) we obtain $y_{S\cap T}^1 = y_{S\cap T}$, which in turn gives $z^1 = y_{S\cap T}$ using (48).
- Using $z^1 = y_{S\cap T}$ in (56) we have that $z^0 = 1 - y_{S\cap T}$.
- Substituting (53) in (34) for $i \in S \cap T$ and using $z^1 = y_{S\cap T}$, we have that $x_i^0 = x_i - y_{S\cap T}$, $\forall i \in S \cap T$.
- Finally, (34) also gives that $x_i^1 = x_i - x_i^0$, $\forall i \in (S\setminus T) \cup (T\setminus S)$.

Applying these substitutions to (34)–(60), we obtain

$$\sum_{i\in S\cap T} x_i - (|S \cap T| - 1) \leq y_{S\cap T}, \tag{61}$$
$$y_{S\cap T} \leq x_i, \qquad \forall i \in S \cap T, \tag{62}$$
$$y_S \leq y_{S\cap T}, \tag{63}$$
$$y_T \leq y_{S\cap T}, \tag{64}$$
$$x_i \leq 1, \qquad \forall i \in S \cap T, \tag{65}$$
$$0 \leq y_S, \tag{66}$$
$$0 \leq y_T, \tag{67}$$
$$0 \leq y_{S\cap T} \leq 1, \tag{68}$$
$$\sum_{i\in S\setminus T} x_i - \sum_{i\in S\setminus T} x_i^0 \leq y_S + y_{S\cap T}(|S\setminus T| - 1), \tag{69}$$
$$\sum_{i\in T\setminus S} x_i - \sum_{i\in T\setminus S} x_i^0 \leq y_T + y_{S\cap T}(|T\setminus S| - 1), \tag{70}$$
$$y_S \leq x_i - x_i^0, \qquad \forall i \in S\setminus T, \tag{71}$$
$$y_T \leq x_i - x_i^0, \qquad \forall i \in T\setminus S, \tag{72}$$
$$y_{S\cap T} \leq 1 - x_i^0, \qquad \forall i \in (S\setminus T) \cup (T\setminus S), \tag{73}$$
$$x_i - x_i^0 \leq y_{S\cap T}, \qquad \forall i \in (S\setminus T) \cup (T\setminus S), \tag{74}$$

$$x_i^0 \leq x_i, \qquad \forall i \in (S\setminus T) \cup (T\setminus S), \tag{75}$$
$$0 \leq x_i^0, \qquad \forall i \in (S\setminus T) \cup (T\setminus S). \tag{76}$$

We will now use the Fourier-Motzkin elimination method to project out all variables x_i^0 from (61)–(76), for $i \in (S\setminus T) \cup (T\setminus S)$, so as to obtain a description of $conv(P^0 \cup P^1)$ in the space of variables $(x_i, y_S, y_T, y_{S\cap T})$.

Notice that constraints (61)–(68) will not play any role in the projection, since they do not involve the variables x_i^0.

Proceeding by induction on the number of eliminated variables, let $I \subseteq S\setminus T$ and $J \subseteq T\setminus S$ be the sets of indices such that variables x_i^0 have been projected out for all $i \in I \cup J$, and let $|I| = p$, $|J| = q$. As induction hypothesis, suppose that after eliminating the variables in $I \cup J$, the formulation is defined by constraints (61)–(68) together with the following inequalities:

$$0 \leq x_i \leq 1, \qquad \forall i \in I \cup J, \tag{77}$$
$$y_S \leq x_i, \qquad \forall i \in I, \tag{78}$$
$$y_T \leq x_i, \qquad \forall i \in J, \tag{79}$$
$$\sum_{i \in S\setminus T} x_i - \sum_{i \in (S\setminus T)\setminus I} x_i^0 \leq y_S + y_{S\cap T}(|S\setminus T| - (p+1)) + p, \tag{80}$$
$$\sum_{i \in T\setminus S} x_i - \sum_{i \in (T\setminus S)\setminus J} x_i^0 \leq y_T + y_{S\cap T}(|T\setminus S| - (q+1)) + q, \tag{81}$$
$$y_S \leq x_i - x_i^0, \qquad \forall i \in (S\setminus T)\setminus I, \tag{82}$$
$$y_T \leq x_i - x_i^0, \qquad \forall i \in (T\setminus S)\setminus J, \tag{83}$$
$$y_{S\cap T} \leq 1 - x_i^0, \qquad \forall i \in ((S\setminus T)\setminus I) \cup ((T\setminus S)\setminus J), \tag{84}$$
$$x_i - x_i^0 \leq y_{S\cap T}, \qquad \forall i \in ((S\setminus T)\setminus I) \cup ((T\setminus S)\setminus J), \tag{85}$$
$$x_i^0 \leq x_i, \qquad \forall i \in ((S\setminus T)\setminus I) \cup ((T\setminus S)\setminus J), \tag{86}$$
$$0 \leq x_i^0, \qquad \forall i \in ((S\setminus T)\setminus I) \cup ((T\setminus S)\setminus J). \tag{87}$$

Note that the induction hypothesis holds when $I = J = \emptyset$ and $p = q = 0$, since (77)–(87) boils down to (69)–(76) in this case. Given I, J, p and q, let us now eliminate variable x_j^0, where $j \in (S\setminus T)\setminus I$, by the Fourier-Motzkin method (the analysis would be similar for $j \in (T\setminus S)\setminus J$). This leads to inequality $x_j \leq 1$ by combining constraints (84) and (85) for j, to inequality $y_S \leq x_j$ by combining (82) and (87) for j, and to inequality $0 \leq x_j$ by combining (86) and (87) for j. Combining (80) and (84) yields

$$\sum_{i \in S\setminus T} x_i - \sum_{i \in (S\setminus T)\setminus(I\cup\{j\})} x_i^0 \leq y_S + y_{S\cap T}(|S\setminus T| - (p+2)) + p + 1.$$

All other combinations of inequalities containing x_j^0 in (77)–(87) lead to redundant constraints. So, clearly, the formulation obtained after projecting out x_j^0 is the same as (77)–(87), with I replaced by $I \cup \{j\}$. This shows that the induction hypothesis holds for all I, J, p, q.

Assume now that we have eliminated all variables x_i^0, $i \in (S \setminus T) \cup (T \setminus S)$. In this case, it follows from the inductive reasoning that constraints (82)–(87) become vacuous. Moreover, the remaining constraints (61)–(68) and (77)–(81) are exactly (22)–(33), the defining constraints of polytope P (except for the bounds $y_S \leq 1$, $y_T \leq 1$ and $x_i \geq 0$, $\forall i \in S \cap T$, which, as stated previously, are among the redundant constraints and can be easily derived from the remaining inequalities). Therefore, we have proved that $P = conv(P^0 \cup P^1)$, which implies that P has integer vertices.

To conclude the proof, we are going to show next that P_{SL}^{2links} is exactly the projection of P on the space of (x, y_S, y_T) variables. Indeed, if we use the Fourier-Motzkin elimination method to project out variable $y_{S \cap T}$ from (22)-(33), then we obtain the standard linearization inequality (6) for S by combining constraints (23) and (26), and for T by combining (23) and (29). Constraints (5) for y_S, y_T, and $i \in S \cap T$ are obtained by combining (22) and (24), and (22) and (27), respectively. Finally, the 2-links (13) and (14) are obtained from inequalities (24), (29) and (26), (27), respectively.

So, we have established that $P_{SL}^{2links} = Proj_{(x, y_S, y_T)}(P)$. Since P is bounded, every vertex of P_{SL}^{2links} is the projection of a vertex of P. This implies that all vertices of P_{SL}^{2links} are integer, since P is integral. Thus, P_{SL}^{2links} is a perfect formulation for X_{SL}, that is, $P_{SL}^{2links} = P_{SL}^*$. □

5 Computational experiments

We have seen in Section 4 that adding all possible 2-links to P_{SL} provides a complete description of P_{SL}^* when the associated function contains two nonlinear monomials. This is not true anymore for functions with three nonlinear monomials. A counterexample is given by the function $f_{3mon}(x) = 5x_1x_2x_4 - 3x_1x_3x_4 - 3x_1x_2x_3 + 2x_3$. If we define P_{SL}^{2links} for f_{3mon} and optimize the corresponding linearized function L_f over P_{SL}^{2links}, we obtain the fractional solution $x_i = 0.5$ for $i = 1, 2, 3, 4$, $y_{134} = 0.5$, $y_{124} = 0$ and $y_{123} = 0.5$.

However, the 2-links might still be helpful for the general case of functions containing more than two nonlinear monomials. In this section, we provide computational evidence showing that the 2-links improve the LP-relaxation of the standard linearization, as well as the computational performance of exact resolution methods. This may not be totally expected, since the 2-links are in relatively small number (quadratic in the number of terms). It appears, however, that capturing relations between pairs of terms improves the standard linearization formulation to a certain extent. (Buchheim and Klein [6] provide results of a related nature for constrained binary quadratic problems, in the sense that they derive valid inequalities for simplified problems involving a single quadratic term, and observe that these inequalities result in significant improvements when applied to the general case.)

In our experiments, we consider two classes of instances of the integer linear program

$$\min \quad L_f(x,y) = \sum_{S\in\mathcal{S}} a_S y_S + \sum_{i\in[n]} a_i x_i \tag{88}$$
$$\text{subject to} \quad (x,y) \in P_{SL} \tag{89}$$
$$x \in \{0,1\}^n. \tag{90}$$

The first class contains RANDOM INSTANCES that are randomly generated in the same way as in [7]. The second class contains so-called VISION INSTANCES; they are inspired by an image restoration problem which is widely studied in the field of computer vision. A description of all instances is provided in the next subsections.

We have used CPLEX 12.6 [1] to run our experiments. We report two types of results. First, we compare the bound obtained when solving the relaxed problem (88)-(89) with the bound obtained when optimizing (88) over P_{SL}^{2links}. Next, we focus on the computational performance of the CPLEX IP-solver when solving the instances to optimality. We compare four different versions of branch & cut to solve (88)-(90), namely:

1. NO CUTS: the automatic cut generation mechanism of CPLEX is disabled to solve the plain standard linearization model (88)-(90).

2. USER CUTS: we solve the standard linearization model enhanced with the addition of 2-links (i.e., over the polytope P_{SL}^{2links}) but without additional automatic cut generation by CPLEX.

3. CPLEX CUTS: the automatic cut generation mechanism of CPLEX is enabled (with the default setting of cut generation parameters) to solve the standard linearization model (88)-(90).

4. CPLEX & USER CUTS (C & U): CPLEX is allowed to use two types of cuts, namely, the 2-links and any additional cuts that it can automatically generate to solve the standard linearization model.

Note that when the 2-link inequalities are used in the branch & cut process, they are treated as a pool of so-called "user cuts". During the process, CPLEX first tries to cut off the current solution by relying on these user cuts only, and next generates its own cuts as needed.

Except for the cut generation parameters, all other IP resolution parameters are set to default. Several preliminary tests have been performed in order to determine the best settings of CPLEX pre-processing parameters. As a result, we chose to set the `Linear Reduction Switch` parameter to the non-default value "perform only linear reductions" since this is the recommended setting by CPLEX whenever there are user cuts. A time limit of 1 hour was set for each instance. All experiments were run on a PC with processor Intel(R) Core(TM) i7-4510U CPU @ 2GHz-2.60GHz, RAM memory of 8 GB, and a Windows 7 64-bit Operating System.

5.1 Random instances

Instance definition. Random instances are generated as in [7]. All functions in this class are to be maximized. They are of two different types.

- SAME-DEGREE. The number of variables n, the number of monomials m and the degree d are given as input. For each triplet (d, n, m), five functions are generated by randomly, uniformly and independently choosing the variables to include in each of the m monomials. All monomials have the same degree d. Their coefficients are drawn uniformly in the interval $[-10, 10]$. All instances in this class have small degree, namely, $d \in \{3, 4\}$.

- RANDOM-DEGREE. n and m are given as an input. Each of the m monomials is generated as follows: first, the degree d of the monomial is chosen from the set $\{2, \ldots, n\}$ with probability 2^{1-d}. In this way, we capture the fact that a random polynomial is likely to have more monomials of lower degree than monomials of higher degree. Then, the variables and coefficient of the monomial are chosen as for the SAME-DEGREE instances. Again, we generate five instances for each pair (n, m). These instances are of much higer degree than the SAME-DEGREE instances. Their average degree will be reported hereunder.

Results. Table 1 presents the results of our experiments on instances RANDOM SAME-DEGREE. Each line displays averages over 5 instances. The first three columns specify parameters d, n, m. The fourth and fifth columns display the relative gaps between the optimal value of the integer programming problem on one hand, and the optimal value of the LP-relaxations of the plain standard linearization (P_{SL}), or of the standard linearization with 2-links (P_{SL}^{2links}) on the other hand. Columns 6 to 9 present the execution times of each of the four tested methods (>3600 is reported whenever no instance was solved to optimality), and columns 10 to 13 give the number of nodes of the branch & cut tree ("–" indicates that no instance was solved to optimality). If the time limit was reached for one or more instances, the unsolved instances are not taken into account in the averages. In addition, we write in parentheses () how many instances were solved to optimality in this case.

Table 1 shows that as a general trend, the addition of 2-links to the standard linearization is useful. Concerning the LP-relaxation bounds, we see that adding the 2-links always improves the bound associated with P_{SL}, by a gap percentage of 0.25% up to 8%. For execution times, it is clear that cuts of any type are helpful, since method NO CUTS is, in most cases, significantly worse than the other methods. In almost all cases, the fastest method is either USER CUTS or CPLEX & USER CUTS (plain CPLEX CUTS is fastest only three times). For large instance sizes, CPLEX & USER CUTS is able to solve more instances than the competing methods. Looking at the number of nodes, it is interesting to notice that even when USER CUTS is the fastest method, it usually generates more nodes than either CPLEX or CPLEX & USER CUTS. This suggests that its performance is due to the smaller amount of time spent in generating the cuts and in solving the corresponding LPs. In contrast, it seems that the performance of CPLEX & USER CUTS is due to the fact that it produces smaller branch & cut trees. It may also be interesting to observe that the difficulty of the problems clearly increases with the density of the instances, that is, with the ratio $\frac{m}{n}$. This observation was also made by Buchheim and Rinaldi [7] (for slightly smaller values of m). Dense instances feature more interactions among monomials. This may increase the intrinsic difficulty of the instances and reduce the effect of adding the 2-links (or other cuts).

Table 1: Results for RANDOM (SAME-DEGREE) instances

Instance			LP bounds: gap %		IP execution times (secs)				IP number of nodes			
d	n	m	P_{SL}	P_{SL}^{2links}	NO CUTS	USER	CPLEX	C & U	NO CUTS	USER	CPLEX	C & U
3	200	500	16.37	12.19	28.00	3.75	9.67	8.24	11557	1208	771	598
3	200	600	27.32	22.80	198.51 (3)	292.46 (4)	416.71	445.25	74409 (3)	93554 (4)	58004	60808
3	200	700	34.96 (2)	28.46 (2)	> 3600 (0)	433.78 (1)	1541.8 (2)	1426.72 (2)	– (0)	104504 (1)	128827 (2)	138958 (2)
3	400	800	4.51	3.49	3.65	2.57	7.46	6.68	423	251	210	135
3	400	900	9.31	7.93	502.41	243.58	104.52	87.75	65848	27489	6481	5405
3	400	1000	14.77 (3)	13.13 (3)	841.36 (1)	434.76 (1)	1334.96 (2)	1884.21 (3)	91939 (1)	37172 (1)	61899 (2)	84172 (3)
3	600	1100	2.78	2.32	14.09	9.88	16.07	14.52	1551	1121	891	626
3	600	1200	6.06	5.37	645.16	333.94	197.13	270.07	46502	25967	8616	12159
3	600	1300	10.17 (3)	9.15 (3)	> 3600 (0)	> 3600 (0)	2157.84 (2)	2234.61 (3)	– (0)	– (0)	84366 (2)	84655 (3)
4	200	350	16.50	11.23	6.50	3.20	9.98	5.89	2218	885	1468	722
4	200	400	22.25	15.84	663.89	207.28	341.68	108.36	262758	64383	70215	26307
4	200	450	28.72	20.81	999.44	324.28	664.39	382.55	285857	81764	98206	49588
4	200	500	35.09 (4)	24.84 (4)	2461.88 (1)	2268.63 (3)	1281.11 (1)	1340.34 (3)	586370 (1)	364125 (3)	143895 (1)	177784 (3)
4	400	550	4.37	3.26	36.97	17.10	14.76	11.6	6753	2743	1806	1318
4	400	600	8.15	5.91	58.79	13.86	63.1	20.19	7416	1458	5563	1184
4	400	650	10.22	7.72	177.74 (4)	681.06	348.79	514.13	22268 (4)	76797	25517	44714
4	400	700	12.25 (3)	8.92 (3)	1343.18 (2)	1179.95 (3)	602.68 (3)	329.05 (3)	130349 (2)	110322 (3)	36622 (3)	21418 (3)
4	600	750	1.54	1.28	3.42	3.05	6.15	5.89	278	234	222	142
4	600	800	2.59	2.14	16.54	12.08	18.37	15.5	1423	940	987	744
4	600	850	5.20	4.02	475.43 (4)	359.65	664.29	316.73	34555 (4)	28255	38502	21381
4	600	900	9.38 (4)	7.59 (4)	103.49 (1)	42.29 (1)	1526.84 (2)	1475.3 (4)	5865 (1)	2183 (1)	63850 (2)	61697 (4)

Table 2 presents the results of our experiments on instances RANDOM RANDOM-DEGREE. The structure of the table is the same as for Table 1 except that d represents now the average degree of the five instances considered in each line.

The interpretation of the results is very similar to the interpretation of Table 1. The 2-link inequalities, by themselves, already improve the LP bound, the execution time and the size of the branch & cut tree, as compared to using no cuts. Method CPLEX CUTS is usually more effective than USER CUTS here. However, CPLEX & USER CUTS still provides an improvement over CPLEX CUTS, both in terms of execution time and size of the enumeration tree, and especially for dense instances. Observe that for this class of instances, we can handle much higher densities $\frac{m}{n}$ than for the SAME-DEGREE instances. This is again similar to the observations in Buchheim and Rinaldi [7], and might be due to the fact that many short monomials (of size 2) tend to appear in this type of instances and may reduce their complexity.

Table 2: Results for RANDOM (RANDOM-DEGREE) instances

Instance			LP bounds: gap %		IP execution times (secs)				IP number of nodes			
d (avg)	*n*	*m*	P_{SL}	P_{SL}^{2links}	NO CUTS	USER	CPLEX	C & U	NO CUTS	USER	CPLEX	C & U
12.6	200	600	12.21	10.15	10.42	8.08	7.15	5.81	5838	3595	398	368
11.2	200	700	12.73	10.73	78.72	30.12	34.74	28.17	35521	12821	3979	2997
11	200	800	18.99	16.10	748.15	254.81	118.55	111.64	257212	76479	10584	9936
13.6	200	900	27.29	23.72	889.37 (2)	690.72 (2)	1029.25	863.39	242729 (2)	135884 (2)	93124	75445
11.2	400	900	3.03	2.43	3.09	1.72	4.15	3.88	859	330	82	61
11	400	1000	3.50	2.82	19.56	6.77	8.87	8.44	4404	1396	286	259
11.4	400	1100	7.27	6.64	55.64 (4)	347.27	59.86	53.66	11289 (4)	61545	2970	2459
11.8	400	1200	7.04 (4)	6.45 (4)	256.80 (3)	117.35 (3)	254.46 (4)	147.80 (4)	42754 (3)	18483 (3)	13987 (4)	9123 (4)
13.8	600	1300	1.38	1.21	2.97	2.53	5.42	5.42	252	207	58	51
11.4	600	1400	3.86	3.57	294.03	238.87	124.30	135.38	36485	27234	5650	5516
12.2	600	1500	4.63	4.10	593.70	228.02	100.28	86.36	67493	24272	3942	3444
12.6	600	1600	5.00	4.53	1374.74 (4)	561.85 (4)	345.37	280.95	110267	47097	11063	8844

5.2 Vision instances

This class of instances is inspired from the image restoration problem, which is widely investigated in computer vision. The problem consists in taking a *blurred image* as an input and in reconstructing an original *sharp base image* based on this input. The interest of the vision instances, beside the practical importance of the underlying problem, is that they have a special structure for which linearization and related pseudo-Boolean optimization methods have proved to perform well (see, e.g., [28], [26], [17], [27]). It is out of the scope of the present paper to work with real-life images: we will rely on a simplified version of the problem and on relatively small scale instances in order to generate structured instances and to evaluate the impact of the 2-link inequalities in this setting. Accordingly, we do not focus on the quality of image restoration (as engineers would typically do), but we devote more attention to the generation of relatively hard instances.

Input image definition. An *image* is a rectangle consisting of $l \times h$ pixels. We model it as a matrix of dimension $l \times h$, where each element represents a pixel which takes value 0 or 1. An input *blurred image* is constructed by considering a *base image* and by applying a perturbation to it, that is, by changing the value of each pixel with a given probability. A base image is denoted as I^{base} and its pixels by p_{ij}^{base}. A blurred image is denoted by I^{blur} and its pixels by p_{ij}^{blur}.

We consider three base images, namely, TOP LEFT RECTANGLE, CENTRE RECTANGLE and CROSS (see Figure 1), with three different sizes 10×10, 10×15 and 15×15.

1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(a) TOP LEFT RECTANGLE

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(b) CENTRE RECTANGLE

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(c) CROSS

Figure 1: Base images: size 10×10

We define three different types of perturbations that can be applied to a base image I^{base} in order to generate I^{blur}, namely:

- NONE: $p_{ij}^{blur} = p_{ij}^{base}$ with probability 1, $\forall (i, j) \in [l] \times [h]$.
- LOW: $p_{ij}^{blur} = p_{ij}^{base}$ with probability 0.95, $\forall (i, j) \in [l] \times [h]$.

- HIGH: $p_{ij}^{blur} = p_{ij}^{base}$ with probability 0.5, $\forall (i, j) \in [l] \times [h]$ with $p_{ij}^{base} = 0$.

Regarding the class HIGH, note that changing the value of every pixel with probability 0.5 would lead to blurred images that are totally unrelated to the original base image; that is why we only apply the perturbation to the "white" pixels (originally taking value $p_{ij}^{base} = 0$) in this case.

Image restoration model. The *image restoration model* associated with a blurred image I^{blur} is defined as an objective function $f(x) = L(x) + P(x)$ that must be minimized. The variables x_{ij}, for all $(i, j) \in [l] \times [h]$, represent the value assigned to each pixel in the output image. $L(x)$ is the linear part and models *similarity* between the input blurred image I_{blur} and the output. $P(x)$ is the nonlinear polynomial part and emphasizes *smoothness*: it aims at taking into account the fact that images typically consist of distinct objects, with pixels inside each object having similar colors, while pixels outside the objects have a different color. Much has been studied on the complex statistics of natural images, but we use here a simplified model.

- *Similarity*: $L(x) = a_L \sum_{i\in[l], j\in[h]} (p_{ij}^{blur} - x_{ij})^2$ minimizes the difference between the value of a pixel in the input image and the value that is assigned to the pixel in the output. Since $x_{ij} \in \{0, 1\}$, $L(x)$ is indeed linear. The coefficient of $L(x)$ is chosen as $a_L = 25$.

- *Smoothness*: $P(x)$ is a polynomial defined by considering 2×2 pixel windows $W_{ij} = \{x_{ij}, x_{i,j+1}, x_{i+1,j}, x_{i+1,j+1}\}$, for $i = 1, \ldots, l - 1$, $j = 1, \ldots, h - 1$. Smoothness is imposed by penalizing the objective function with a nonlinear monomial for each window W_{ij}. The more the assignment of variables in the window W_{ij} looks like a checkerboard, the higher the coefficient of the monomial, thus giving preference to smoother assignments. Table 3 provides the penalties used for each of the 16 assignments of values to a 2×2 window. So for example, the assignment of values $x_{ij} = x_{i,j+1} = 1$, $x_{i+1,j} = x_{i+1,j+1} = 0$ (third row in Table 3) gives rise to the monomial $30x_{ij}x_{i,j+1}(1 - x_{i+1,j})(1 - x_{i+1,j+1})$ in the objective function.

The choice of coefficients in Table 3 and of the linear coefficient a_L was made by running a series of preliminary calibration tests aimed at finding a good balance between the importance given to smoothness and to similarity, so that the resulting instances are not too easy to solve.

Instance definition. For each image size in $\{10 \times 10, 10 \times 15, 15 \times 15\}$ and for each base image, we have generated five instances, namely: one sharp image (the

Table 3: Variable assignments in 2×2 windows, and associated penalty coefficients

Variable assignments								Coefficient
0 0	1 1							10
0 0	1 1							
0 0	0 0	0 1	1 0	1 1	1 1	1 0	0 1	20
0 1	1 0	0 0	0 0	1 0	0 1	1 1	1 1	
1 1	0 0	1 0	0 1					30
0 0	1 1	1 0	0 1					
1 0	0 1							40
0 1	1 0							

base image with perturbation type NONE), two blurred images with perturbation type LOW, and two blurred images with perturbation type HIGH.

Notice that the difference between the five instances associated with a given size and a given base image is due to the input blurred image, which results from a random perturbation. This only affects the similarity term $L(x)$, while the smoothness model $P(x)$ remains the same for all instances of a given size.

Results. Tables 4, 5 and 6 report the results obtained for images of size 10×10, 10×15 and 15×15, respectively. The structure of the tables is the same as for random instances, except for the first two columns, which respectively specify the base image and the perturbation applied. For the perturbation type NONE, we report the result obtained for a single instance. For the perturbation type LOW or HIGH, we report the averages for two instances.

We can see that, in all cases, the bounds derived from P_{SL} are very bad (ranging from 400% to 2000% above the optimal value). The bounds are significantly improved (by about 50%) when we add 2-links to the formulation (see column P_{SL}^{2links}), but they still remain very weak. Concerning execution times, methods NO CUTS and USER CUTS perform poorly and reach the time limit for almost every instance. A drastic improvement in computing times is achieved by CPLEX CUTS, which solves the easiest instances in just a few seconds and the most difficult ones in 110 seconds at most. Interestingly, however, a further significant improvement is obtained by CPLEX & USER CUTS, which solves all instances in less than 13 seconds. CPLEX & USER CUTS is in some cases up to ten times faster than CPLEX CUTS and always solves the problem at the root node, which suggests that its excellent performance is indeed due to the addition of the 2-links.

It is interesting to notice the major effect played by the structure of the instances. Indeed, vision instances have much worse LP gaps than random instances, and are much more dense (reaching $n = 225$ variables and $m = 1598$ terms for the 15×15 images). For the vision instances, we observe dramatic differences among the four solution methods that we have tested. Nevertheless, these instances turn out to be much easier to solve to optimality than random instances: it appears that the cuts generated by CPLEX and the 2-link inequalities are very complementary and provide remarkable benefits for the class of vision instances. Of course, the larger the size of the image, the more difficult the problem becomes. Perturbation types also have a big influence on complexity, since HIGH perturbation type instances are always harder to solve, as one might expect. Finally, the choice of base images does not seem to have any impact on the difficulty of the instances.

Table 4: Results for VISION instances of size 10×10

Instance (10×10)		LP bounds: gap %		IP execution times (secs)				IP number of nodes			
Base image	Perturbation	P_{SL}	P_{SL}^{2links}	NO CUTS	USER	CPLEX	C & U	NO CUTS	USER	CPLEX	C & U
TOP LEFT RECT	NONE	584.07	296.70	> 3600	61.31	2.75	4.76	–	122037	0	0
TOP LEFT RECT	LOW	679.57	352.33	> 3600	105.91	4.70	0.74	–	220003	4	0
TOP LEFT RECT	HIGH	482.95	253.18	> 3600	> 3600	16.22	2.52	–	–	77.5	0
CENTRE RECT	NONE	1074.53	581.13	> 3600	304.89	6.05	0.81	–	625644	0	0
CENTRE RECT	LOW	1038.39	562.50	> 3600	494.95	7.41	0.94	–	1027936	0	0
CENTRE RECT	HIGH	525.25	277.48	> 3600	> 3600	11.44	1.48	–	–	0	0
CROSS	NONE	1989.29	1100	> 3600	206.25	3.25	0.95	–	418973	0	0
CROSS	LOW	1679.44	931.69	> 3600	669.79	8.49	1.63	–	1407712	0	0
CROSS	HIGH	379.48	192	> 3600	3062.91 (1)	10.15	1.45	–	5727483 (1)	3.5	0

Table 5: Results for VISION instances of size 10×15

Instance (10×15)		LP bound: gap %		IP execution times (secs)				IP number of nodes			
Base image	Perturbation	P_{SL}	P_{SL}^{2links}	NO CUTS	USER	CPLEX	C & U	NO CUTS	USER	CPLEX	C & U
TOP LEFT RECT	NONE	621.80	318.05	> 3600	> 3600	6.22	1.98	–	–	0	0
TOP LEFT RECT	LOW	749.58	396.66	> 3600	> 3600	15.50	2.04	–	–	3.5	0
TOP LEFT RECT	HIGH	480.87	251.87	> 3600	> 3600	38.49	3.35	–	–	42.5	0
CENTRE RECT	NONE	859.13	458.65	> 3600	> 3600	7.94	2.04	–	–	0	0
CENTRE RECT	LOW	1015.13	552.04	> 3600	> 3600	15.74	2.59	–	–	3.5	0
CENTRE RECT	HIGH	464.31	242.59	> 3600	> 3600	49.42	3.11	–	–	64.5	0
CROSS	NONE	1608.33	883.33	> 3600	> 3600	32.37	2.26	–	–	0	0
CROSS	LOW	1790.63	999.23	> 3600	> 3600	20.78	2.54	–	–	7.5	0
CROSS	HIGH	468.24	245.07	> 3600	> 3600	38.22	3.46	–	–	38.5	0

Table 6: Results for vision instances of size 15×15

Instance (15×15)		LP bounds (gap %)		IP execution times (secs)				IP number of nodes			
Base image	Perturbation	P_{SL}	P_{SL}^{2links}	no cuts	user	cplex	c & u	no cuts	user	cplex	c & u
top left rect	none	660.90	340.26	> 3600	> 3600	19.5	3.49	–	–	0	0
top left rect	low	714.29	374.27	> 3600	> 3600	28.06	6.41	–	–	0	0
top left rect	high	565.72	302.48	> 3600	> 3600	111.3	12.86	–	–	126.5	0
centre rect	none	698.13	366.75	> 3600	> 3600	30.12	4.71	–	–	0	0
centre rect	low	851.09	457.40	> 3600	> 3600	38.33	8.44	–	–	6.5	0
centre rect	high	483.33	253.69	> 3600	> 3600	97.17	10.34	–	–	222	0
cross	none	1284.52	698.57	> 3600	> 3600	16.54	5.63	–	–	0	0
cross	low	1457.22	801.10	> 3600	> 3600	22.30	7.26	–	–	0	0
cross	high	530.46	282.23	> 3600	> 3600	103.75	11.02	–	–	80	0

6 Conclusions

In this paper, we have provided new results on the standard linearization technique, a well-known approach to the optimization of multilinear polynomials in binary variables. We have introduced the 2-link inequalities, a set of valid inequalities that express a relation between pairs of monomials, and that strengthen the LP-relaxation of the standard linearization. Our main result is that, for a function containing at most two nonlinear terms, the 2-links, together with the classical standard linearization inequalities, provide a perfect formulation of the standard linearization polytope P^*_{SL}.

For the general case of objective functions with more than two nonlinear terms, the 2-links are not enough to obtain a complete description of the standard linearization polytope. However, our computational experiments show that the 2-links are still helpful for various classes of instances. On one hand, the 2-links always improve the LP-relaxation bounds derived from the standard linearization. The improvement is much larger for the computer vision instances, which have very bad standard linearization bounds to begin with, than for unstructured random instances. On the other hand, our results show that 2-links can be very effective within a branch & cut framework. This is especially true when solving vision instances, where the addition of 2-links to the pool of available cuts allows CPLEX to obtain the optimal solution without any branching and, as a consequence, significantly reduces the solution time. The magnitude of this effect is even more surprising given that the 2-links are rather simple inequalities and that they are in relatively small number (quadratic in the number of terms of the objective function).

There are many interesting open questions arising from our research. Of course, it is unlikely to obtain a complete description of the standard linearization polytope in the general case (unless $\mathcal{P} = \mathcal{NP}$). It remains however interesting to investigate whether there are other special cases of functions for which the 2-links provide a complete description of P^*_{SL}. A related question is to identify specially structured instances for which the impact of the 2-links is computationally significant, as is the case for our vision instances. Finally, another natural question is whether it is possible to generate similar inequalities by establishing a link between three or more monomials, and whether these inequalities would further tighten the lower bounds and improve computational performance.

Acknowledgements

We thank the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office (grant P7/36) for funding this project. We also thank Endre

Boros for several discussions on the topic of our research and on the generation of computer vision instances, Quentin Louveaux and Christoph Buchheim for their suggestions on the 2-monomial case and on the setting of our computational experiments, and Anja Fischer for her insights on the case of nested monomials.

References

[1] IBM ILOG CPLEX Optimizer 12.6. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[2] Al-Khayyal, F.A., Falk, J.E.: Jointly constrained biconvex programming. Mathematics of Operations Research **8**(2), 273–286 (1983)

[3] Balas, E.: Disjunctive programming: properties of the convex hull of feasible points. GSIA Management Science Research Report MSRR 348, Carnegie Mellon University (1974). Published as invited paper in Discrete Applied Mathematics 89 (1998) 3–44

[4] Balas, E.: Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. SIAM Journal on Algebraic Discrete Methods **6**(3), 466–486 (1985)

[5] Boros, E., Hammer, P.L.: Pseudo-Boolean optimization. Discrete Applied Mathematics **123**(1), 155–225 (2002)

[6] Buchheim, C., Klein, L.: Combinatorial optimization with one quadratic term: spanning trees and forests. Discrete Applied Mathematics **177**, 34–52 (2014)

[7] Buchheim, C., Rinaldi, G.: Efficient reduction of polynomial zero-one optimization to the quadratic case. SIAM Journal on Optimization **18**(4), 1398–1413 (2007)

[8] Burer, S., Letchford, A.N.: Non-convex mixed-integer nonlinear programming: a survey. Surveys in Operations Research and Management Science **17**(2), 97–106 (2012)

[9] Conforti, M., Cornuéjols, G., Zambelli, G.: Integer Programming, *Graduate Texts in Mathematics*, vol. 271. Springer, Switzerland (2014)

[10] Crama, Y.: Concave extensions for nonlinear 0–1 maximization problems. Mathematical Programming **61**(1), 53–60 (1993)

[11] Crama, Y., Hammer, P.L.: Boolean Functions: Theory, Algorithms, and Applications. Cambridge University Press, New York, N. Y. (2011)

[12] D'Ambrosio, C., Lodi, A.: Mixed integer nonlinear programming tools: a practical overview. 4OR **9**(4), 329–349 (2011)

[13] De Simone, C.: The cut polytope and the boolean quadric polytope. Discrete Mathematics **79**(1), 71–75 (1990)

[14] Del Pia, A., Khajavirad, A.: A polyhedral study of binary polynomial programs. To appear in Mathematics of Operations Research, May 2016

[15] Djeumou Fomeni, F., Kaparis, K., Letchford, A.: Cutting planes for RLT relaxations of mixed 0-1 polynomial programs. Mathematical Programming **151**(2), 639–658 (2015)

[16] Fischer, A., Fischer, F., McCormick, S.T.: Matroid optimisation problems with nested non-linear monomials in the objective function. NAM Preprint 2016-2, Institute for Numerical and Applied Mathematics, University of Goettingen, March 2016

[17] Fix, A., Gruber, A., Boros, E., Zabih, R.: A hypergraph-based reduction for higher-order binary Markov random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(7), 1387–1395 (2015)

[18] Fortet, R.: L'algèbre de Boole et ses applications en recherche opérationelle. Cahiers du Centre d'Études de Recherche Opérationelle **4**, 5–36 (1959)

[19] Fortet, R.: Applications de l'algèbre de Boole en recherche opérationelle. Revue Française d'Automatique, Informatique et Recherche Opérationelle **4**(14), 17–26 (1960)

[20] Glover, F., Woolsey, E.: Further reduction of zero-one polynomial programming problems to zero-one linear programming problems. Operations Research **21**(1), 156–161 (1973)

[21] Glover, F., Woolsey, E.: Technical note: converting the 0-1 polynomial programming problem to a 0-1 linear program. Operations Research **22**(1), 180–182 (1974)

[22] Hammer, P.L., Rosenberg, I., Rudeanu, S.: On the determination of the minima of pseudo-Boolean functions. Studii si Cercetari Matematice **14**, 359–364 (1963). In Romanian

[23] Hammer, P.L., Rudeanu, S.: Boolean Methods in Operations Research and Related Areas. Springer-Verlag, Berlin (1968)

[24] Hansen, P., Jaumard, B., Mathon, V.: State-of-the-art survey: constrained nonlinear 0–1 programming. ORSA Journal on Computing **5**(2), 97–119 (1993)

[25] Hemmecke, R., Köppe, M., Lee, J., Weismantel, R.: Nonlinear integer programming. In: 50 Years of Integer Programming 1958-2008, pp. 561–618. Springer (2010)

[26] Ishikawa, H.: Transformation of general binary MRF minimization to the first-order case. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(6), 1234–1249 (2011)

[27] Kappes, J.H., Andres, B., Hamprecht, F.A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B.X., Kröger, T., Lellmann, J., Komodakis, N., Savchynskyy, B., Rother, C.: A comparative study of modern inference techniques for structured discrete energy minimization problems. International Journal of Computer Vision **115**(2), 155–184 (2015)

[28] Kolmogorov, V., Rother, C.: Minimizing nonsubmodular functions with graph cuts – a review. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(7), 1274–1279 (2007)

[29] Luedtke, J., Namazifar, M., Linderoth, J.: Some results on the strength of relaxations of multilinear functions. Mathematical Programming **136**(2), 325–351 (2012)

[30] McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. Mathematical Programming **10**(1), 147–175 (1976)

[31] Ryoo, H.S., Sahinidis, N.V.: Analysis of bounds for multilinear functions. Journal of Global Optimization **19**(4), 403–424 (2001)

[32] Watters, L.J.: Reduction of integer polynomial programming problems to zero-one linear programming problems. Operations Research **15**, 1171–1174 (1967)

[33] Zangwill, W.I.: Media selection by decision programming. Journal of Advertising Research **5**, 30–36 (1965)