MIT Open Access Articles

Embedding Formulations and Complexity for Unions of Polyhedra

Embedding Formulations and Complexity for Unions of Polyhedra

Juan Pablo Vielma

Sloan School of Management, Massachusetts Institute of Technology, jvielma@mit.edu

It is well known that selecting a good Mixed Integer Programming (MIP) formulation is crucial for an effective solution with state-of-the art solvers. While best practices and guidelines for constructing good formulations abound, there is rarely a systematic construction leading to the best possible formulation. We introduce embedding formulations and complexity as a new MIP formulation paradigm for systematically constructing formulations for disjunctive constraints that are optimal with respect to size. More specifically, they yield the smallest possible ideal formulation (i.e. one whose LP relaxation has integral extreme points) among all formulations that only use 0-1 auxiliary variables. We use the paradigm to characterize optimal formulations for SOS2 constraints and certain piecewise linear functions of two variables. We also show that the resulting formulations can provide a significant computational advantage over all known formulations for piecewise linear functions.

Key words: integer programming; mixed-integer programming; mixed-integer modeling; strong
   formulation; piecewise linear
MSC2000 subject classification:
OR/MS subject classification: Primary: ; secondary:
History: 29-Apr-2016, 25-Jan-2017, 23-Apr-2017, 18-May-2017

1. Introduction

In its more than 50 years of history Mixed Integer Programming (MIP) has become an indispensable tool in Operations Research and Management Science. Enormous strides have been made in the theoretical and computational issues arising in solving MIP problems and commercial MIP solvers can solve a wide range of problems (Jünger et al. 2010). One of the reasons for the success of MIP is its modeling flexibility. For instance, 0-1 MIP can be used to model disjunctive constraints (i.e. the selection over a finite number of alternatives) appearing in a wide range of applications in transportation (Croxton et al. 2003, Roberti et al. 2014), telecommunication (D'Andreagiovanni et al. 2013) and scheduling (Manne 1960, Pinedo 2012). Formulating problems using MIP is often straightforward. However, as most textbooks warn, some care should be taken in constructing MIP formulations as some formulation attributes can severely affect the effectiveness of solvers.

Good formulations can be obtained following simple guidelines, but more elaborate techniques can provide a significant computational advantage (Vielma 2015). For instance, consider the classical *Special Ordered Sets of Type 2 (SOS2)* introduced by Beale and Tomlin (1970). SOS2 constraints on variables $\lambda \in \mathbb{R}^{n+1}$ require that (i) $\sum_{i=1}^{n+1} \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i \in \{1, \ldots, n+1\}$, and (ii) at most two λ_i variables can be non-zero at the same time and if $\lambda_i > 0$ and $\lambda_j > 0$, then they must be adjacent variables (i.e. $|i - j| \leq 1$). A textbook formulation for SOS2 constraints for $n = 4$ is given by

$$\sum_{j=1}^{5} \lambda_j = 1, \quad \lambda_j \geq 0 \quad \forall j \in \{1, \ldots, 5\}, \quad \sum_{i=1}^{4} y_i = 1, \quad y \in \{0,1\}^4 \tag{1a}$$

$$\lambda_1 \leq y_1, \quad \lambda_2 \leq y_1 + y_2, \quad \lambda_3 \leq y_2 + y_3, \quad \lambda_4 \leq y_3 + y_4, \quad \lambda_5 \leq y_4. \tag{1b}$$

To evaluate this formulation we consider two formulation attributes: formulation size and strength. For size we use the standard measure corresponding to the smallest number of linear inequalities needed to describe the formulation (where we count an equation as two inequalities). Formulation strength can be evaluated in many ways (Vielma 2015, Section 2.2); we choose to check if the formulation satisfies the following property.

DEFINITION 1 (IDEAL FORMULATION). Let $Ax + By \leq b, \quad y \in \mathbb{Z}^k$ be a MIP formulation[†]. The Linear Programming (LP) relaxation of this formulation is the polyhedron described by $Ax + By \leq b$, which for simplicity we assume has at least one extreme point or basic feasible solution[‡]. We say the MIP formulation is *ideal* if and only if all the extreme points of its LP relaxation satisfy the integrality constraints $y \in \mathbb{Z}^k$.

Ideal formulations are in a sense *strongest possible* (Vielma 2015, Section 2.2) and usually outperform similarly sized non-ideal formulations. Formulation (1) fares well with regards to size, as its version for general n only requires $2n + 6$ inequalities, with 4 of these coming from equations (the bounds on the y variables are not actually needed). However, this small size comes at the cost of the formulation not being ideal. Indeed, for (1) to be ideal every extreme point (λ, y) of its LP relaxation must satisfy $y \in \mathbb{Z}^4$. The LP relaxation of (1) is obtained by replaxing $y \in \{0,1\}^4$ by $0 \leq y_i \leq 1$ for all $i \in \{1, \ldots, 4\}$ and we can check that $\lambda = (1/2, 1/2, 0, 0, 0)$ and $y = (1/2, 0, 1/2, 0)$ is an extreme point of this resulting set. Fortunately, Padberg (2000) showed how to strengthen (1) to an ideal formulation without increasing the total number of inequalities. For $n = 4$ the resulting formulation is given by

$$\sum_{j=1}^{5} \lambda_j = 1, \quad \lambda_1 \geq 0, \quad \lambda_5 \geq 0, \quad \sum_{i=1}^{4} y_i = 1, \quad y \in \{0,1\}^4 \tag{2a}$$

$$\lambda_1 \leq y_1 \leq \lambda_1 + \lambda_2 \leq y_1 + y_2 \leq \lambda_1 + \lambda_2 + \lambda_3 \leq y_1 + y_2 + y_3 \leq \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4. \tag{2b}$$

[†]Where A, B and b are appropriately sized rational matrices and vector.

[‡]For a case in which this assumption is relaxed see Vielma (2017).

While both formulations have nearly the same size, the majority of the inequalities of (2) correspond to general inequalities (2b), which are not just variable bounds. In contrast, only half the inequalities of (1) are general inequalities (see Table 1 and Corollary 2). General inequalities usually have a stronger computational impact than variable bounds as the later can be treated implicitly by LP and MIP solvers. Hence the larger number of general inequalities of (2) can cancel its advantage from being ideal and indeed it is often computationally outperformed by non-ideal formulation (1). A solution to this issue can be found in an advanced formulation technique introduced by Vielma and Nemhauser (2011). For $n = 4$ this technique yields the formulation for SOS2 given by

$$\sum_{j=1}^{5} \lambda_j = 1, \quad \lambda_j \geq 0 \quad \forall j \in \{1, \ldots, 5\}, \quad y \in \{0, 1\}^2 \tag{3a}$$

$$\lambda_1 + \lambda_5 \leq 1 - y_1, \quad \lambda_3 \leq y_1, \quad \lambda_4 + \lambda_5 \leq 1 - y_2, \quad \lambda_1 + \lambda_2 \leq y_2. \tag{3b}$$

This formulation is also ideal, but its version for general n requires around $2\lceil \log_2 n \rceil + n + 2$ inequalities and only $2\lceil \log_2 n \rceil + 2$ of these are general inequalities (see Corollary 3). This allows formulation (3) to have a significant computational advantage over (1), (2) and all known formulations for SOS2 (Vielma et al. 2010, Vielma and Nemhauser 2011). An even more dramatic issue arises if we consider a 2-dimensional generalization of SOS2 constraints used to model piecewise linear functions of two variables (Lee and Wilson 2001, Vielma et al. 2010). As detailed in Table 1, the cost of going from the non-ideal 2-dimensional generalization of (1) to the ideal 2-dimensional generalization of (2) is a significant increase in the number of inequalities, particularly general inequalities. However, the 2-dimensional generalization of formulation (3) still has a linear number of inequalities and a logarithmic number of general inequalities. This again gives it a significant computational advantage over all known formulations for piecewise linear functions of two variables (Vielma et al. 2010, Vielma and Nemhauser 2011).

The only disadvantage of formulation (3) is its increase in complexity. In particular, while for formulations (1) and (2) we can easily interpret the role of the 0-1 variables ($y_i = 1$ if and only if λ_i and λ_{i+1} can be non-zero at the same time), the role of the 0-1 variables is not so clear for (3) (for instance it uses two 0-1 variables instead of four). This increase in complexity makes it hard to generalize (3) to other constraints. In fact, the 2-dimensional generalization of (3) only works for very specific piecewise linear functions, while the 2-dimensional generalizations of (1) and (2) work for a wide range of piecewise linear functions.

In this paper we propose a new MIP formulation paradigm that should allow extending the success of formulation (3) to a wide range of applications. In particular, this paradigm can construct ideal formulations for any disjunctive constraint that requires a set of variables to be in the union of

Formulation	Traditional SOS2		2-D Generalization	
	General Inequalities	Bounds	General Inequalities	Bounds
(1)	$n+1$	$n+1$	$\left(\sqrt{n/2}+1\right)^2$	$\left(\sqrt{n/2}+1\right)^2$
(2)	$2(n-1)$	2	$\binom{2\sqrt{n/2}}{\sqrt{n/2}}$	$\left(\sqrt{n/2}+1\right)^2$
(3)	$2\lceil\log_2 n\rceil$	$[n-1,n+1]$	$4\left\lceil\log_2\sqrt{n/2}\right\rceil+2$	$\left(\sqrt{n/2}+1\right)^2$

Table 1 Sizes of Formulations for SOS2 Constraints and its 2-dimensional Generalization. We omit the number of inequalities from equations, which is 4 for (1) and (2), and 2 for (3).

a finite number of polyhedra with mild technical requirements. We denote the formulations obtained through this paradigm *embedding formulations* as they are based on a geometric construction that *embeds* the disjunctive constraints into a higher dimensional space that contains both the original variables in the constraint (e.g. the λ variables for SOS2 constraints) and the 0-1 variables of the formulation (e.g. the y variables). One characteristic of embedding formulations is allowing a flexible use of 0-1 variables that include both traditional uses such as in (1) and (2), and more complex uses such as in (3). This flexibility is the key to replicating the success of (3) as we show that the size of an embedding formulations can be extremely sensitive to the specific use of 0-1 variables. For this reason we also study the size of the smallest embedding formulation for a disjunctive constraint when we consider all possible uses of 0-1 variables. We denote this the embedding complexity of the associated union of polyhedra. This complexity measure has theoretical interest on its own, but can also be used to evaluate the potential for improvement of existing formulations. For instance, studying this complexity allows us to show that (3) is (nearly) optimal with regard to size. To the best of our knowledge, this result is the first lower bound on sizes of *mixed* integer formulations and one of the few results for any class of integer programming formulations. The only other similar results we are aware of are the techniques to show lower bounds on sizes of combinatorial and *pure* integer formulations introduced by Kaibel and Weltge (2015), Weltge (2015). Finally, we show how the embedding formulation paradigm can be used to generalize the 2-dimensional version of (3) to a wider range of piecewise linear functions than what was considered in Vielma and Nemhauser (2011). We also show how the resulting formulations can significantly outperform all other known formulations for piecewise linear functions of two variables. This generalization is based on the computational calculation of a convex hull associated to the embedding formulation. To the best of our knowledge, this is the first example of a computational construction of an effective MIP formulation.

Throughout the paper we use the following notation. For a set $S \subseteq \mathbb{R}^d$ we let $\operatorname{conv}(S)$, $\operatorname{aff}(S)$, $\operatorname{span}(S)$ and $\dim(S)$ be the convex hull, affine hull, linear span and the dimension of S respectively. For a polyhedron $P \subseteq \mathbb{R}^d$ we let $\operatorname{ext}(P)$ and $\operatorname{ray}(P)$ be the set of extreme points and extreme rays of P. We also let P_∞ be the recession cone of P. Given two vectors $a, b \in \mathbb{R}^V$ for a finite index set V we let $a \cdot b = \sum_{v \in V} a_v b_v$ be the inner product between a and b. We also let $\mathbf{0} \in \mathbb{R}^V$ be the vector of all zeros and $\mathbf{e}^v \in \mathbb{R}^V$ be the unit vector such that $\mathbf{e}^v_u = 1$ if $u = v$ and $\mathbf{e}^v_u = 0$ otherwise (The index set V is often evident form the context so we omit it in this notation). Finally, we let $[\![n]\!] := \{1, \ldots, n\}$, $[\![a, b]\!] := \{a, a+1, \ldots, b-1, b\}$ and $\mathbb{Q}$ be the set of rational numbers.

2. Geometric Construction of Formulations

We consider MIP formulations for the disjunctive constraint

$$x \in \bigcup_{i=1}^n P^i \tag{4}$$

where $\mathcal{P} := (P^i)_{i=1}^n$ is a finite family of polyhedra in $\mathbb{R}^d$ that satisfy the following assumption.

ASSUMPTION 1. *The family of polyhedra $\mathcal{P} := (P^i)_{i=1}^n$ is such that P^i is a rational polyhedron and $\operatorname{ext}(P^i) \neq \emptyset$ for all $i \in [\![n]\!]$, and $P^i_\infty = P^j_\infty$ for all $i, j \in [\![n]\!]$.*

Disjunctive constraint (4) is exactly the type of practical[§] constraints that can be modeled using 0-1 MIP (Jeroslow and Lowe 1984) and MIP formulations for it can be constructed using a wide range of techniques (Vielma 2015). For instance, if the polyhedra are described by linear inequalities, the following result from Jeroslow and Lowe (1984), Balas (1985) gives an ideal and small formulation for (4).

THEOREM 1. *Let $\mathcal{P} := (P^i)_{i=1}^n$ be a finite family of polyhedra in $\mathbb{R}^d$ satisfying Assumption 1. Furthermore, for each $i \in [\![n]\!]$, let $A^i \in \mathbb{Q}^{m_i \times d}$ and $b^i \in \mathbb{Q}^{m_i}$ be such that $P^i = \{x \in \mathbb{R}^d : A^i x \leq b^i\}$. Then an ideal formulation of (4) is given by*

$$A^i x^i \leq b^i y_i \quad \forall i \in [\![n]\!], \quad x = \sum_{i=1}^n x^i, \quad \sum_{i=1}^n y_i = 1, \quad y \in \{0, 1\}^n. \tag{5}$$

Formulation (5) achieves the feat of being ideal and small through the use of continuous auxiliary variables $x^i \in \mathbb{R}^d$, which copy the original variables x for each polyhedron. However, these continuous auxiliary variables can take away the potential computational advantage of (5), particularly when small and ideal formulations without the continuous auxiliary variables are available (e.g. formulation (3) for SOS2 constraints). Unfortunately, there is no known general formulation for disjunctive constraints that is small, ideal and does not use these continuous auxiliary variables.

[§]We can also consider unions of polyhedra without extreme points, but this is uncommon in practice.

To remedy this, we present a general geometric procedure to construct formulations without continuous auxiliary variables that is always ideal and can sometimes yield small formulations. The basic steps of this procedure are depicted in Figure 1 for the polyhedra P^1 and P^2 in the left of the figure. The first step embeds the polyhedra into a space that contains an additional 0-1 variable y_1 by converting the disjunction from $S := P^1 \cup P^2$ to $S^+ := (P^1 \times \{1\}) \cup (P^2 \times \{0\})$. The second step takes the convex hull of this embedding to obtain $Q = \text{conv}(S^+)$. By construction we have that $S^+ = Q \cap (\mathbb{R}^2 \times \mathbb{Z})$ and hence the projection of $Q \cap (\mathbb{R}^2 \times \mathbb{Z})$ onto the x variables is equal to S. Furthermore, by construction we also have that the extreme points of Q have an integral y_1 component. Hence, an ideal formulation of S is given by $(x, y_1) \in Q = \text{conv}(S^+)$ and $y_1 \in \mathbb{Z}$. Finally, because Q is a rational polyhedron, this formulation contains only linear inequalities with rational coefficients and the integrality constraint on y_1.
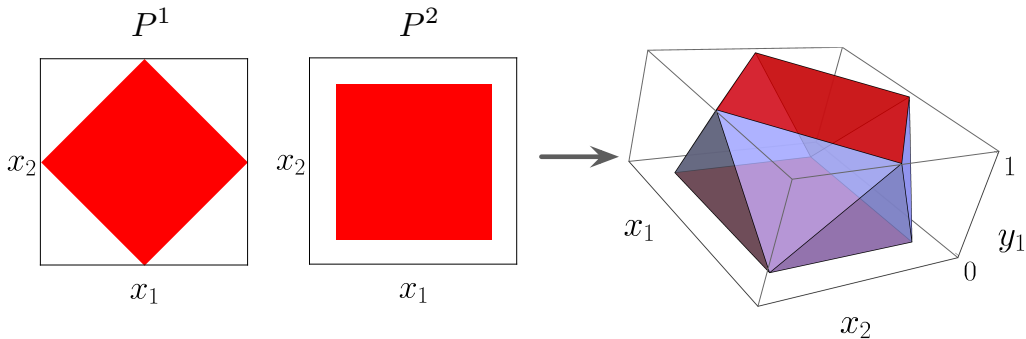
Figure 1   Two polyhedra (left in red), their embedding into a space with one 0-1 variable (right in red) and the convex hull of this embedding (right in light blue).

2.1. Embedding Formulations

One possible generalization of the embedding procedure in Figure 1 to more than two polyhedra is to embed $\bigcup_{i=1}^{n} P^i$ into $\bigcup_{i=1}^{n} (P^i \times \{\mathbf{e}^i\})$ where $\mathbf{e}^i$ is the i-th unit vector. This procedure is known as the *Cayley trick* or *Cayley Embedding* and is used to study Minkowski sums of polyhedra (e.g. Huber et al. (2000), Karavelas et al. (2013), Weibel (2007)). We here consider a further generalization that pairs the polyhedra P^i with any set of pairwise disjoint 0-1 vectors instead of just the unit vectors $\mathbf{e}^i$. The following proposition and its proof shows that the convex hull of the resulting embedding is a rational polyhedron that can be used to construct an ideal formulation of $x \in \bigcup_{i=1}^{n} P^i$.

PROPOSITION 1. *Let $\mathcal{P} := (P^i)_{i=1}^{n}$ be a finite family of polyhedra in $\mathbb{R}^d$ satisfying Assumption 1, $k \geq \lceil \log_2 n \rceil$, $H := (h^i)_{i=1}^{n} \in \mathcal{H}_k(n) := \left\{ (h^i)_{i=1}^{n} : h^i \in \{0,1\}^k \quad \forall i \in [\![n]\!], \quad h^i \neq h^j \quad \forall i \neq j \right\}$ be a family of pairwise distinct 0-1 vectors indexed in the same way as the polyhedra and*

$$Q(\mathcal{P}, H) := \text{conv}\left(\bigcup_{i=1}^{n} P^i \times \{h^i\} \right). \tag{6}$$

Then

(i) $Q(\mathcal{P}, H)$ is a rational polyhedron,

(ii) $(x, y) \in Q(\mathcal{P}, H) \cap (\mathbb{R}^d \times \mathbb{Z}^k) \quad \Leftrightarrow \quad \exists i \in [\![n]\!] \ s.t. \ y = h^i \wedge x \in P^i, \ and$

(iii) $\text{ext}(Q(\mathcal{P}, H)) \subseteq \mathbb{R}^d \times \{0, 1\}^k$.

Proof. By Assumption 1, $\{P^i \times \{h^i\}\}_{i=1}^n$ is a finite family of non-empty polyhedra with identical recession cones. Then, by Lemma 4.41 and Corollary 4.44 in Conforti et al. (2014), $Q(\mathcal{P}, H)$ is a rational polyhedron and $Q(\mathcal{P}, H) = \text{conv}\left(\bigcup_{i=1}^n \bigcup_{v \in \text{ext}(P^i)} \{v \times h^i\}\right) + \text{cone}\left(\bigcup_{r \in \text{ray}(P^1)} \{r \times \mathbf{0}\}\right)$. This shows the first two properties of $Q(\mathcal{P}, H)$. The last follows by additionally noting that the h^i's are distinct extreme points of $[0, 1]^k$. $\quad \square$

Similarly to the example depicted in Figure 1 we obtain a formulation from $Q(\mathcal{P}, H)$ by interpreting H as possible values of a k-dimensional 0-1 variable y.

COROLLARY 1 (**Embedding Formulation**). *Let $\mathcal{P} := (P^i)_{i=1}^n$ be a finite family of polyhedra in $\mathbb{R}^d$ satisfying Assumption 1, $k \geq \lceil \log_2 n \rceil$ and $H \in \mathcal{H}_k(n)$. Then an ideal formulation of $x \in \bigcup_{i=1}^n P^i$ is given by*

$$(x, y) \in Q(\mathcal{P}, H), \quad y \in \mathbb{Z}^k. \tag{7}$$

Furthermore, besides the integrality constraints on y, this formulation only includes linear inequalities with rational coefficients. We refer to (7) as the embedding formulation of $\mathcal{P}$ associated to H, to H as the encoding of the formulation and to $Q(\mathcal{P}, H)$ as the LP relaxation of the formulation.

Proof. The formulation is valid because of property (ii) in Proposition 1, includes only linear inequalities with rational coefficients because of property (i) in Proposition 1, and is ideal because of property (iii) in Proposition 1. $\quad \square$

Encoding H can have a strong impact in the size of $Q(\mathcal{P}, H)$, but different encodings can yield formulations of the same size or even equivalent polyhedra $Q(\mathcal{P}, H)$. The following straightforward lemma shows one such possible equivalency.

LEMMA 1. *Let $\mathcal{P} := (P^i)_{i=1}^n$ be a finite family of polyhedra in $\mathbb{R}^d$ satisfying Assumption 1, $k_1, k_2 \geq \lceil \log_2 n \rceil$, $H \in \mathcal{H}_{k_1}(n)$ and $G \in \mathcal{H}_{k_2}(n)$. If there exists an affine map $A : \mathbb{R}^{k_1} \to \mathbb{R}^{k_2}$ such that A is a bijection between $\text{conv}(H)$ and $\text{conv}(G)$ then $Q(\mathcal{P}, H)$ is affinely isomorphic to $Q(\mathcal{P}, G)$.*

In particular, if $H := (h^i)_{i=1}^n$ where $h^i = \mathbf{e}^i$ for all $i \in [\![n]\!]$, and $G := \{g^i\}_{i=1}^n$ where $g^i = \mathbf{e}^{\pi(i)}$ for all $i \in [\![n]\!]$ and $\pi : [\![n]\!] \to [\![n]\!]$ is a permutation, then

$$(x, y) \in Q(\mathcal{P}, G) \quad \Leftrightarrow \quad (x, y^\pi) \in Q(\mathcal{P}, H)$$

where $y_{\pi(i)}^\pi = y_i$ for all $i \in [\![n]\!]$.

Lemma 1 shows that when the encoding uses n unit vectors $\mathbf{e}^i \in \{0,1\}^n$ the specific order or polytope-vector assignment in the embedding is inconsequential. Hence, when an encoding of this form is used we assume the unit vectors are assigned in their natural order and refer to the resulting embedding formulation as the *unary encoded* formulation as this encoding can be interpreted as a *unary encoding* of the selection among the polyhedra. A completely different class of encodings are obtained when n is a power of 2 and $(h^i)_{i=1}^n = \{0,1\}^k$ for $k = \log_2 n$. This case can be interpreted as a *binary encoding* of the selection among the polyhedra and corresponds to the encodings with the smallest number of components or *bits*. For this reason we refer to embedding formulations resulting from such encodings as *binary encoded* formulations. Unlike unary encoded formulations, in Section 3 we show that permuting the order of a binary encoding can lead to binary encoded formulations of significantly different sizes. This potential size variability over binary and other encodings motivates the following complexity measure for unions of polyhedra, which quantifies the size of its smallest embedding formulation.

Definition 2 (Embedding Complexity). For a polyhedron Q let $\text{size}(Q)$ be equal to the minimum number of inequalities needed to describe Q (equations are counted as two inequalities). Then, for a family of polyhedra $\mathcal{P} := (P^i)_{i=1}^n$ satisfying Assumption 1 we let its *embedding complexity* be

$$\text{mc}(\mathcal{P}) := \min \left\{ \text{size}(Q(\mathcal{P}, H)) : H \in \bigcup\nolimits_{k \geq \lceil \log_2 n \rceil} \mathcal{H}_k(n) \right\}.$$

Constructing an embedding formulation even for a fixed H requires a potentially costly convex hull calculation. Calculating the embedding complexity has the added difficulty of minimizing the size of these convex hulls over all possible encodings. Fortunately, as we show in the following sections it is sometimes possible to give tight bounds on the embedding complexity of specially structured disjunctions such as SOS2 constraints. We also show how we can computationally construct embedding formulations for piecewise linear functions that can provide a significant computational advantage. In both these cases the unions of polyhedra considered have a special combinatorial structure (specifically, all polyhedra are faces of a fixed simplex). For an example of how embedding formulations can lead to a computational advantage for disjunctions without this special structure we refer the reader to Huchette et al. (2016).

3. Bounds on embedding complexity for SOS2 constraints

SOS2 constraints can be posed as a disjunctive constraint of the form (4) as follows.

Definition 3. Let $\Delta^{n+1} := \left\{ \lambda \in \mathbb{R}_+^{n+1} : \sum_{i=1}^{n+1} \lambda_j = 1 \right\}$. A family of polyhedra $\mathcal{P} := (P^i)_{i=1}^n$ in $\mathbb{R}^{n+1}$ is the special ordered sets of type 2 or SOS2 constraint on Δ^{n+1} if and only if

$$P^i := \left\{ \lambda \in \Delta^{n+1} : \lambda_j \leq 0 \quad \forall j \notin \{i, i+1\} \right\} = \text{conv}\left(\{ \mathbf{e}^i, \mathbf{e}^{i+1} \} \right).$$

To fully characterize embedding formulations for SOS2 constraints we need to consider the possibility of an encoding not being full dimensional (i.e. $H \in \mathcal{H}_k(n)$ and $\dim(H) < k$), which we handle through the following definition.

DEFINITION 4. For $H := (h^i)_{i=1}^n$, let $L(H) := \mathrm{aff}(H) - h^1$ be the linear space parallel to the affine hull of H.

The following proposition shows that the size and structure of the inequalities of $Q(\mathcal{P}, H)$ have a relatively simple description for SOS2 constraints.

PROPOSITION 2. Let $\mathcal{P} := (P^i)_{i=1}^n$ be the SOS2 constraint on Δ^{n+1}, $H := (h^i)_{i=1}^n \in \mathcal{H}_k(n)$, $c^i = h^{i+1} - h^i$ for $i \in [\![n-1]\!]$ and for $b \in L(H) \setminus \{\mathbf{0}\}$ let $M(b) := \{y \in L(H) : b \cdot y = 0\}$ be the linear (or central) hyperplane defined by b in $L(H)$. Finally, let $\{b^l\}_{l=1}^L \subseteq L(H) \setminus \{\mathbf{0}\}$ be such that $\{M(b^l)\}_{l=1}^L$ is the set of linear hyperplanes spanned by $(c^i)_{i=1}^{n-1}$ in $L(H)$ and $J = \{1, n+1\} \cup \left\{j \in [\![2, n]\!] : L(H) = \mathrm{span}\left(\{c^i\}_{i \in [\![n-1]\!] \setminus \{j-1\}}\right)\right\}$. Then $(\lambda, y) \in Q(\mathcal{P}, H)$ if and only if

$$\sum_{j=1}^{n+1} \lambda_j = 1, \quad y \in \mathrm{aff}(H) \tag{8a}$$

$$\sum_{j=1}^{n+1} \min\left\{b^l \cdot h^j, b^l \cdot h^{j-1}\right\} \lambda_j \le b^l \cdot y \le \sum_{j=1}^{n+1} \max\left\{b^l \cdot h^j, b^l \cdot h^{j-1}\right\} \lambda_j \quad \forall l \in [\![L]\!] \tag{8b}$$

$$\lambda_j \ge 0 \qquad \forall j \in J, \tag{8c}$$

where we let $h^0 = h^1$ and $h^{n+1} = h^n$. Furthermore, $\dim(Q(\mathcal{P}, H)) = n + \dim(H)$, equations (8a) precisely describe $\mathrm{aff}(Q(\mathcal{P}, H))$ and all inequalities in (8b) and (8c) are facet defining for $Q(\mathcal{P}, H)$. Finally, none of the facets defined by (8b) can be defined by a variable bound on λ.

The proof of Proposition 2 is slightly technical and follows from a complete characterization of the facial structure of $Q(\mathcal{P}, H)$ for SOS2 constraints, so we postpone it to Section 5.1.

One notable characteristic of (8) is the simple form of the inequalities. However, this simplicity hides the possibility of discovering rather complicated formulations whose validity is not straightforward. Indeed, combining Proposition 2 and Corollary 1 we have that a valid (and ideal) formulation for SOS2 constraints is (λ, y) satisfy (8) and $y \in \mathbb{Z}^k$. We can easily check that $(\lambda, y) = (\mathbf{e}^j, h^i)$ is feasible for this formulation if $j \in \{i, i+1\}$. The converse can sometimes be easily verified for encodings with a specific structure, but is not evident from generic formulation (8). To complicate matters further, this converse does not directly imply infeasibility of $(\lambda, y) = (1/2)(\mathbf{e}^j, h^i) + (1/2)(\mathbf{e}^l, h^i)$ for $j \in \{i, i+1\}$ and $l \notin \{i, i+1\}$, which is also required for validity of the formulation. In Section 5.2 give an example of how even for fixed n and H it may not be straightforward to verify validity of the formulation based on (8) *directly* (i.e. without using Corollary 1 and the fact that (8) describes $Q(\mathcal{P}, H)$). This example illustrates how the embedding formulation procedure can be useful to discover or construct unexpected formulations whose validity is not immediately evident.

Besides constructing unexpected formulations, Proposition 2 allows us to recover existing formulations and determine that one of these is the smallest embedding formulation for SOS2 constraints. Characterization (8) of $Q(\mathcal{P}, H)$ considers two classes of facets: those defined by variable bounds (8c) and those defined by more general inequalities (8b). As noted in the introduction the computational cost of variable bounds tends to be smaller so we refine our notion of formulation size to consider this in our analysis.

DEFINITION 5. Let $\mathcal{P} := (P^i)_{i=1}^n$ be the SOS2 constraint on Δ^{n+1} and $H := (h^i)_{i=1}^n \in \mathcal{H}_k(n)$. We let $\mathrm{size}_G(Q(\mathcal{P}, H))$ and $\mathrm{size}_B(Q(\mathcal{P}, H))$ be the number of inequalities in (8b) and (8c) respectively so that $\mathrm{size}(Q(\mathcal{P}, H)) = \mathrm{size}_G(Q(\mathcal{P}, H)) + \mathrm{size}_B(Q(\mathcal{P}, H)) + 2(1 + k - \dim(H))$. Finally, we let $\mathrm{mc}_G(\mathcal{P}) := \min\left\{\mathrm{size}_G(Q(\mathcal{P}, H)) : H \in \bigcup_{k \geq \lceil \log_2 n \rceil} \mathcal{H}_k(n)\right\}$.

The first formulation we can recover with Proposition 2 is the ideal formulation introduced in Padberg (2000) that was illustrated in (2).

COROLLARY 2. *Let $\mathcal{P} := (P^i)_{i=1}^n$ be the SOS2 constraint on Δ^{n+1} and H be the unary encoding, then $(\lambda, y) \in Q(\mathcal{P}, H)$ if and only if*

$$\sum_{j=1}^{n+1} \lambda_j = 1, \quad \sum_{i=1}^n y_i = 1, \quad y \in \{0,1\}^n, \tag{9a}$$

$$\sum_{j=l+2}^{n+1} \lambda_j \leq \sum_{i=l+1}^n y_i, \quad \sum_{j=1}^l \lambda_j \leq \sum_{i=1}^l y_i \qquad \forall l \in [\![n-1]\!] \tag{9b}$$

$$\lambda_1 \geq 0, \quad \lambda_{n+1} \geq 0. \tag{9c}$$

Furthermore, all inequalities in (9b)–(9c) are facet defining and hence $\mathrm{size}(Q(\mathcal{P}, H)) = 2n + 4$, $\mathrm{size}_G(Q(\mathcal{P}, H)) = 2(n-1)$ and $\mathrm{size}_B(Q(\mathcal{P}, H)) = 2$.

Proof. For the unary encoding we have $c^i = \mathbf{e}^{i+1} - \mathbf{e}^i$ for all $i \in [\![n-1]\!]$ and hence $L(H) = \left\{y \in \mathbb{R}^n : \sum_{j=1}^n y_j = 0\right\}$. Furthermore, the set of hyperplanes spanned by $(c^i)_{i=1}^{n-1}$ is $\left\{M(b^l)\right\}_{l=1}^L$ for $L = n - 1$ and $b^l = (n-l)\sum_{i=1}^l \mathbf{e}^i - l\sum_{i=l+1}^n \mathbf{e}^i$ for each $l \in [\![L]\!]$. Finally, the elements of $(c^i)_{i=1}^{n-1}$ are linearly independent and hence $J = \{1, n+1\}$ and (8) for the unary encoding becomes

$$\sum_{j=1}^{n+1} \lambda_j = 1, \quad \sum_{i=1}^n y_i = 1, \quad y \in \{0,1\}^n \tag{10a}$$

$$(n-l)\sum_{j=1}^l \lambda_j - l\sum_{j=l+1}^{n+1} \lambda_j \leq (n-l)\sum_{i=1}^l y_i - l\sum_{i=l+1}^n y_i \quad \forall l \in [\![n-1]\!] \tag{10b}$$

$$(n-l)\sum_{i=1}^l y_i - l\sum_{i=l+1}^n y_i \leq (n-l)\sum_{j=1}^{l+1} \lambda_j - l\sum_{j=l+2}^{n+1} \lambda_j \quad \forall l \in [\![n-1]\!]. \tag{10c}$$

$$\lambda_1 \geq 0, \quad \lambda_{n+1} \geq 0. \tag{10d}$$

If we subtract $n - l$ times the implied equation $\sum_{i=1}^n y_i = \sum_{j=1}^{n+1} \lambda_j$ from (10c) and divide by n, and add l times this same implied equation to (10b) and divide by n we have that (10) is equivalent to (9). $\square$

We can also use Proposition 2 to recover and generalize the logarithmic formulation from Vielma and Nemhauser (2011) that we illustrated in (3). For that we need the following special class of binary encodings with adjacent elements (in the order induced by the SOS2 constraints) that only differ in one bit or coordinate.

DEFINITION 6. We say $H = (h^i)_{i=1}^n \in \mathcal{H}_{\lceil \log_2 n \rceil}(n)$ is a *gray code* if and only if for all $i \in [\![n-1]\!]$ we have $\sum_{j=1}^{\lceil \log_2 n \rceil} \left| h_j^i - h_j^{i+1} \right| = 1$.

The following corollary shows that formulation (8) for a gray code is precisely the logarithmic formulation when n is a power of two and otherwise eliminates some redundancy of the logarithmic formulation (see Muldoon (2012) for an alternate derivation and discussion about this redundancy).

COROLLARY 3. *Let* $\mathcal{P} := (P^i)_{i=1}^n$ *be the SOS2 constraint on* Δ^{n+1} *and* H *be a gray code, then* $(\lambda, y) \in Q(\mathcal{P}, H)$ *if and only if*

$$\sum_{j=1}^{n+1} \lambda_j = 1 \tag{11a}$$

$$\sum_{j=1}^{n+1} \min\left\{ h_l^j, h_l^{j-1} \right\} \lambda_j \le y_l \le \sum_{j=1}^{n+1} \max\left\{ h_l^j, h_l^{j-1} \right\} \lambda_j \quad \forall l \in [\![\lceil \log_2 n \rceil]\!] \tag{11b}$$

$$\lambda_j \ge 0 \qquad \forall j \in [\![n+1]\!]. \tag{11c}$$

where we let $h^0 = h^1$ *and* $h^{n+1} = h^n$. *Furthermore, all inequalities in* (11b) *are facet defining and hence* $\mathrm{size}_G(Q(\mathcal{P}, H)) = 2 \lceil \log_2 n \rceil$. *In addition, at least* $n - 1$ *of the inequalities in* (11c) *are facet defining and hence* $|\mathrm{size}_B(Q(\mathcal{P}, H)) - n| \le 1$ *and* $|\mathrm{size}(Q(\mathcal{P}, H)) - (2 \lceil \log_2 n \rceil + n + 2)| \le 1$. *Finally, these last two bounds are tight.*

Proof. For any gray code $c^i \in \bigcup_{j=1}^{\lceil \log_2 n \rceil} \{-\mathbf{e}^j, \mathbf{e}^j\}$ for each $i \in [\![n-1]\!]$ and $L(H) = \mathbb{R}^{\lceil \log_2 n \rceil}$. Hence, the set of hyperplanes spanned by $(c^i)_{i=1}^{n-1}$ is $\left\{ M(b^l) \right\}_{l=1}^L$ for $L = \lceil \log_2 n \rceil$ and $b^l = \mathbf{e}^i$ for each $l \in [\![L]\!]$. This yields the results concerning non-bound inequalities.

For the bound on the number of facet defining inequalities in (11c) note that $L(H) \ne \mathrm{span}\left(\{c^i\}_{i \in [\![n-1]\!] \setminus \{j-1\}} \right)$ if and only if there exist $l \in [\![\lceil \log_2 n \rceil]\!]$ such that $|c^{j-1}| = \mathbf{e}^l$ and $|c^i| \ne \mathbf{e}^l$ for all $i \in [\![n-1]\!] \setminus \{j-1\}$. Then, the number of inequalities $\lambda_j \ge 0$ which fail to be facet defining is the number of bits that only change once in H, which is given by

$$T_n^1(H) := \left| \left\{ l \in [\![\lceil \log_2 n \rceil]\!] : \exists i_0 \in [\![n-1]\!] \text{ s.t. } h_l^{i_0} \ne h_l^{i_0+1} \text{ and } h_l^i = h_l^{i+1} \quad \forall l \in [\![n-1]\!] \setminus \{i_0\} \right\} \right|.$$

To show that $T_n^1(H) \le 2$ assume for a contradiction that $n \ge 3$ and $T_n^1(H) \ge 3$. Without loss of generality we may assume there exist $0 = i_0 < i_1 < i_2 < i_3 < i_4 = n$ such that for all $l \in [\![3]\!]$ we have $h_l^i = 1$ for all $i \in [\![i_l]\!]$ and $h_l^i = 0$ for all $i \in [\![i_l+1, n]\!]$. Let $k = \lceil \log_2 n \rceil$ be the dimension or number of bits of H so that $n \ge 2^{k-1} + 1$. Then, because the indices i_1, i_2, i_3 divide H into $4 = 2^2$ sets, there exits $l \in [\![4]\!]$ such that $i_l - i_{l-1} \ge \left\lceil (2^{k-1} + 1)/4 \right\rceil = 2^{k-3} + 1$. However, because $h^i \ne h^j$ and $h_l^i = h_l^j$ for

all $l \in [\![3]\!]$ and $i,j \in [\![i_{l-1}+1, i_l]\!]$, we must have $2^{k-3} \geq i_l - i_{l-1} \geq 2^{k-3}+1$ which yields a contradiction. For the tightness of the bounds note that for $H = ((0,0)^T, (1,0)^T, (1,1)^T)$ we have $T_n^1(H) = 2$ and for $H = ((1,0,0)^T, (1,1,0)^T, (0,1,0)^T, (0,1,1)^T, (1,1,1)^T, (1,0,1)^T, (0,0,1)^T, (0,0,0)^T)$ we have $T_n^1(H) = 0$. Finally, we recover the logarithmic formulation when n is a power of two (e.g. formulation (9) in Vielma and Nemhauser (2011)) by subtracting (11a) from the second inequality in (11b), noting that $\min\{h_l^j, h_l^{j-1}\} = 1$ if $h_l^j = h_l^{j-1} = 1$ and $\min\{h_l^j, h_l^{j-1}\} = 0$ otherwise, and noting that $1 - \max\{h_l^j, h_l^{j-1}\} = 1$ if $h_l^j = h_l^{j-1} = 0$ and $1 - \max\{h_l^j, h_l^{j-1}\} = 0$ otherwise. $\quad\square$

Finally, we can use Proposition 2 to give tight bounds on the embedding complexity of SOS2 constraints, which show that the logarithmic formulation from Corollary 3 is (nearly) optimal.

PROPOSITION 3. *Let* $\mathcal{P} := (P^i)_{i=1}^n$ *be the SOS2 constraint on* Δ^{n+1}, *then*

- $\mathrm{mc}_G(\mathcal{P}) = 2\lceil \log_2 n \rceil$, *and*
- $n + 3 + \lceil \log_2 n \rceil \leq \mathrm{mc}(\mathcal{P}) \leq n + 3 + 2\lceil \log_2 n \rceil$.

Proof. Let $\{b^l\}_{l=1}^L \subseteq L(H) \setminus \{\mathbf{0}\}$ be such that $\{M(b^l)\}_{l=1}^L$ is the set of linear hyperplanes spanned by $\{c^i\}_{i=1}^{n-1}$ in $L(H)$. By Proposition 2 an easy upper bound on the number of variable bounds $\lambda_j \geq 0$ that are not facet defining is L, which is achieved precisely if for all $l \in [\![L]\!]$ there is only one $i \in [\![n-1]\!]$ such that $b^l \cdot c^i \neq 0$. Hence we have $\mathrm{size}(Q(\mathcal{P}, H)) \geq 2 + 2L + n + 1 - L = n + 3 + L$. Then, to obtain a lower bound on both $\mathrm{mc}_G(\mathcal{P})$ and $\mathrm{mc}(\mathcal{P})$ we need to minimize L. For that we note that L is equal to the number of 1-flats of the (central) hyperplane arrangement $\{\{b \in L(H) : c^i \cdot b = 0\}\}_{i=1}^{n-1}$ in $L(H)$. Because $\mathrm{span}\left(\{c^i\}_{i=1}^{n-1}\right) = L(H)$ the number of such 1-flats is at least $\dim(L(H)) = \dim(H)$. Because all elements of H are pairwise distinct we have that $L \geq \dim(H) \geq \lceil \log_2 n \rceil$, which yields the lower bounds. The upper bounds follow from Corollary 3. $\square$

The smallest size embedding formulation is achieved through the use of a very specific class of binary encoding. In the following section we explore how rare this class is with respect to yielding small embedding formulations.

3.1. Size distribution for binary encodings

If $n = 2^k$ for some $k \in \mathbb{Z}$, $\mathcal{P} := (P^i)_{i=1}^n$ is the SOS2 constraint and $H \in \mathcal{H}_k(n)$, then Proposition 2 shows that the number of facets of $Q(\mathcal{P}, H)$ defined by general inequalities (8b) is upper bounded by $2\binom{n-1}{k-1}$. The following proposition suggests that this upper bound may be nearly achieved. We include a proof of this result in Section 5.3.

PROPOSITION 4. *Let* $n = 2^k$ *for some* $k \in \mathbb{Z}$ *and* $\mathcal{P} := (P^i)_{i=1}^n$ *be the SOS2 constraint. There exist* $H \in \mathcal{H}_k(n)$ *such that* $\mathrm{size}_G(Q(\mathcal{P}, H))$ *is equal to twice the number of affine hyperplanes spanned by* $\{0,1\}^{k-1}$.

It is believed that the number of affine hyperplanes spanned by $\{0,1\}^{k-1}$ is close to its trivial upper bound of $\binom{n/2}{k-1}$ for $n = 2^k$ (e.g. Aichholzer and Aurenhammer (1996)). Both this upper bound and the general-inequality-defined facet bound of $2\binom{n-1}{k-1}$ grow roughly as $n^{\log_2 n}$¶, which suggests that the worst case of $\text{size}_G\left(Q\left(\mathcal{P}, H\right)\right)$ for a binary encoded formulation is quasi-polynomial in n. Hence, it seems like an unfortunate selection of the specific binary encoding can lead to a formulation that is significantly larger than the lower bound from Proposition 3 or even the size of the unary encoded formulation from Corollary 2. Because of its link with the number of hyperplanes spanned by subsets of $\{0,1\}^{k-1}$ (or $\{-1,0,1\}^{k-1}$), understanding the typical size of a binary encoded embedding formulation for SOS2 constraints may prove extremely challenging (e.g. Voigt and Ziegler (2006)). For this reason we only pursue a simple empirical study of the distribution of sizes for these formulations. For this study we selected $k \in [\![3,6]\!]$ and calculated $\text{size}_G\left(Q\left(\mathcal{P}, H\right)\right)$ for randomly selected binary encodings (the ones associated to a random permutation of $\{0,1\}^k$). For $k = 3$ we considered all $40,320$ possible encodings, while for $k \in \{4,5\}$ we only used a random sample of $10,000$ encodings and for $k = 6$ we only used a random sample of $1,000$ encodings (calculating the formulation sizes for $k = 6$ was already computational intensive). The results of this study are presented in Figure 2. The figure presents histograms for $\text{size}_G\left(Q\left(\mathcal{P}, H\right)\right)$ with random
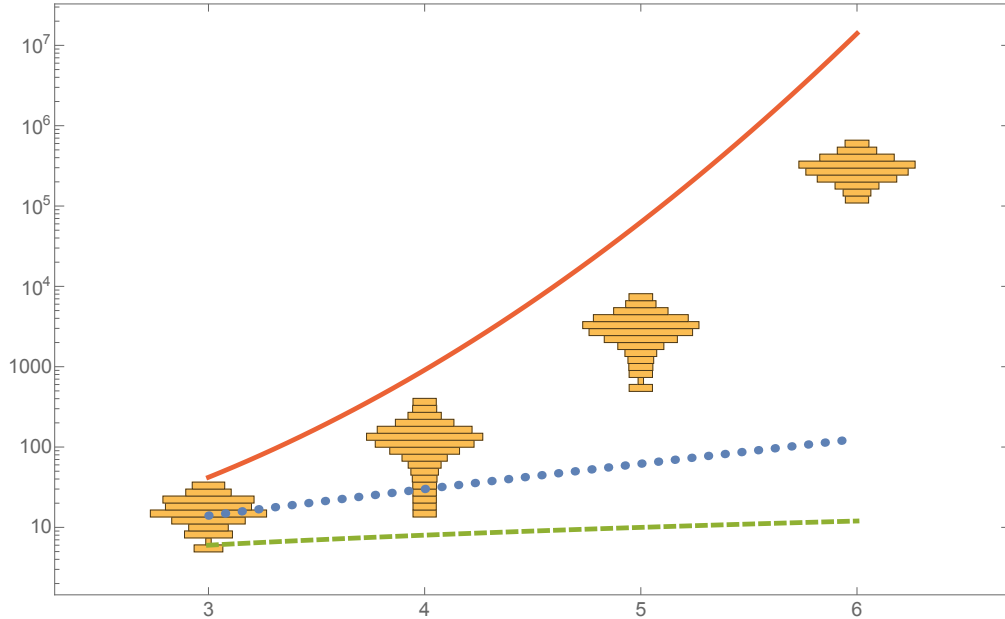
Figure 2 $\text{size}_G\left(Q\left(\mathcal{P}, H\right)\right)$ **for** $k \in [\![3,6]\!]$ **and randomly selected binary encodings.**
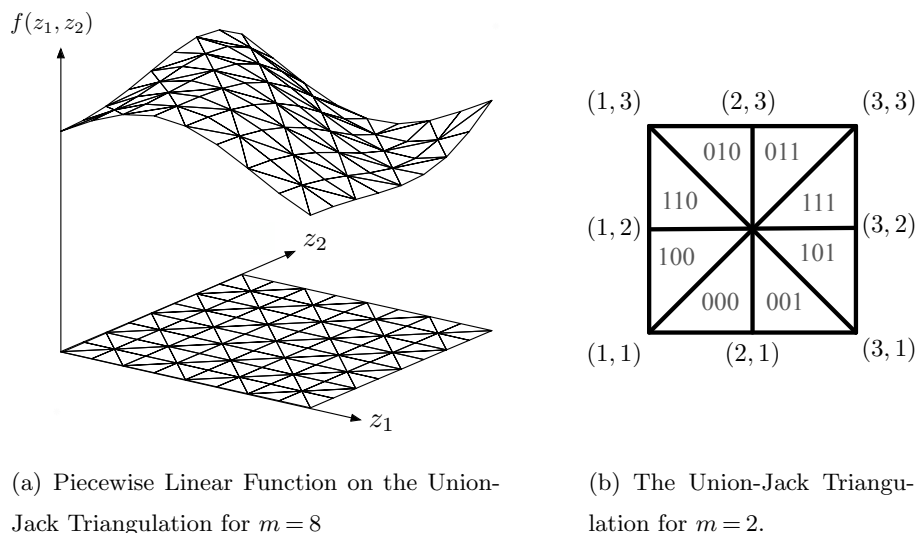
binary encoding H for each k, together with the trivial upper bound of $2\binom{n-1}{k-1}$ (depicted by the

¶We have $\Omega\left(n^{(1-\varepsilon)\log_2 n}\right) = \binom{n/2}{k-1} \le \binom{n-1}{k-1} \le n^{\log_2 n}$ for all $\varepsilon > 0$.

solid red line), $\text{size}_G\left(Q\left(\mathcal{P},H\right)\right)$ for the unary encoding (depicted by the dotted blue line) and $\text{size}_G\left(Q\left(\mathcal{P},H\right)\right)$ for the optimal binary encoded formulation (depicted by the dashed green line). The figure shows that the typical value of $\text{size}_G\left(Q\left(\mathcal{P},H\right)\right)$ for a binary encoding seems to be much closer to the upper bound and suggests that a randomly selected encoding may often lead to a formulation that is significantly larger than even the unary encoded formulation. Hence a careful encoding selection appears crucial to obtain a small formulation.

4. Formulations for Piecewise Linear Functions of Two Variables

The results in Section 3.1 show that it may be hard to construct small embedding formulations. However, we now show how small embedding formulations can be constructed for multivariate piecewise linear functions. MIP formulations for multivariate piecewise linear functions can be constructed using standard generalizations of SOS2 constraints (e.g. Lee and Wilson (2001), Vielma and Nemhauser (2011), Vielma et al. (2010)). For simplicity, we only consider formulations for piecewise linear functions of two variables defined on grid triangulations on $[\![m+1]\!]^2$ such as those depicted in Figure 3. More precisely, we consider functions $f : [1, m+1]^2 \to \mathbb{R}$ that are continuous in $[1, m+1]^2$ and affine in each triangle of the triangulation (e.g. for the triangulation in Figure 3(b) it is affine in $\text{conv}\left(\{(1,1),(1,2),(2,2)\}\right)$).

(a) Piecewise Linear Function on the Union-Jack Triangulation for $m = 8$

(b) The Union-Jack Triangulation for $m = 2$.

Figure 3   **Piecewise Linear Functions of Two Variables and Grid Triangulations.**

The following proposition summarizes the standard generalization of SOS2 constraints used to give a disjunctive representation of the graph of a piecewise linear function. It also describes how an embedding formulation can be used to model the corresponding disjunctive constraint and yield a formulation for the graph of this function.

PROPOSITION 5. *Let* $\mathcal{T} = \left\{ T_{u,v}^1, T_{u,v}^2 \right\}_{u,v=1}^m$ *be a grid triangulation on* $[\![m+1]\!]^2$ *so that for all* $u, v \in [\![m]\!]$ *we have* $T_{u,v}^1 \cup T_{u,v}^2 = \{u, u+1\} \times \{v, v+1\}$, $\left| T_{u,v}^1 \right| = \left| T_{u,v}^2 \right| = 3$ *and either* $T_{u,v}^1 \cap T_{u,v}^2 = \{(u,v), (u+1,v+1)\}$ *or* $T_{u,v}^1 \cap T_{u,v}^2 = \{(u+1,v), (u,v+1)\}$ *(e.g. for the triangulation in Figure 3(b) we can take* $T_{2,1}^1 = \{(2,1), (3,1), (2,2)\}$ *and* $T_{2,1}^2 = \{(3,1), (3,2), (2,2)\}$ *or exchange* $T_{2,1}^1$ *and* $T_{2,1}^2$*). Finally, let* $\Delta_2^{m+1} := \left\{ \lambda \in \mathbb{R}_+^{[\![m+1]\!]^2} : \sum_{u,v=1}^{m+1} \lambda_{(u,v)} = 1 \right\}$, $n = 2m^2$ *and* $\mathcal{P}(\mathcal{T}) := (P^i)_{i=1}^n$ *so that* $P^{2(m(u-1)+(v-1))+t} = P\left(T_{u,v}^t \right) := \left\{ \lambda \in \Delta_2^{m+1} : \lambda_{(\bar{u},\bar{v})} \leq 0 \quad \forall (\bar{u},\bar{v}) \notin T_{u,v}^t \right\}$ *for all* $u, v \in [\![m]\!]$ *and* $t \in \{1, 2\}$.

Then for any continuous function $f : [1, m+1] \to \mathbb{R}$ *that is affine in* $\mathrm{conv}(T)$ *for each* $T \in \mathcal{T}$, *we have that a disjunctive representation of its graph* $\mathrm{gr}(f) := \{(x, z) \in \mathbb{R}^3 : f(x) = z\}$ *is given by*

$$\sum_{u,v=1}^{m+1} u\lambda_{(u,v)} = x_1, \quad \sum_{u,v=1}^{m+1} v\lambda_{(u,v)} = x_2, \quad \sum_{u,v=1}^{m+1} f(u,v)\lambda_{(u,v)} = z \tag{12a}$$

$$\lambda \in \bigcup_{i=1}^n P^i. \tag{12b}$$

If $k \geq \lceil \log_2 n \rceil$ *and* $H \in \mathcal{H}_k(n)$ *then an ideal formulation of* $\mathrm{gr}(f)$ *is given by* (12a), $(\lambda, y) \in Q(\mathcal{P}(\mathcal{T}), H)$ *and* $y \in \mathbb{Z}^k$.

Constructing and analyzing embedding formulations for triangulations can be significantly more complicated than for SOS2 constraints. However, the analysis can be partially achieved through the following simple lemma that shows how a description of $Q(\mathcal{P}(\mathcal{T}), H)$ can be obtained from an ideal formulation of (12b) whose binary variables are *compatible* with the encoding H. That is if the formulation can be re-interpreted as a formulation of $(\lambda, y) \in \bigcup_{i=1}^n P_i \times \{h^i\}$ (the LP relaxation of such formulation corresponds to Q in the lemma).

LEMMA 2. *Let* $\mathcal{P} := (P^i)_{i=1}^n$ *be a finite family of polyhedra in* $\mathbb{R}^d$ *satisfying Assumption 1,* $k \geq \lceil \log_2 n \rceil$, $H := (h^i)_{i=1}^n \in \mathcal{H}_k(n)$ *and* $Q \subseteq \mathbb{R}^{d+k}$ *be a rational polyhedron. If* $\mathrm{ext}(Q) \subseteq \mathbb{R}^d \times \mathbb{Z}^k$ *and*

$$(x, y) \in Q \cap \left(\mathbb{R}^d \times \mathbb{Z}^k \right) \quad \Leftrightarrow \quad \exists i \in [\![n]\!] \text{ s.t. } y = h^i \wedge x \in P^i, \tag{13}$$

then $Q = Q(\mathcal{P}, H)$.

Proof. By (13), the definition of $Q(\mathcal{P}, H)$ and convexity of Q we have $Q(\mathcal{P}, H) \subseteq Q$. Furthermore, (13) and the assumption on $\mathrm{ext}(Q)$ yield $\mathrm{ext}(Q) \subseteq \bigcup_{i=1}^n P^i \times \{h^i\}$. By Proposition 1 and the fundamental theorem of integer programming applied to $Q \cap (\mathbb{R}^d \times \mathbb{Z}^k)$ and $Q(\mathcal{P}, H) \cap (\mathbb{R}^d \times \mathbb{Z}^k)$ we further have $Q_\infty = P_\infty^1 \times \{\mathbf{0}\} = Q(\mathcal{P}, H)_\infty$. Finally, combining these properties with Minkowski-Weyl we get $Q = \mathrm{conv}(\mathrm{ext}(Q)) + Q_\infty \subseteq Q(\mathcal{P}, H)$, which shows the result. $\square$

Combining Lemma 2 and formulations from Lee and Wilson (2001) and Vielma and Nemhauser (2011) we can give a rather precise analysis for a triangulation known as the *union-jack* (Todd 1977), which is depicted in Figure 3(b) (See Vielma and Nemhauser (2011) for a precise description).

In particular, we have that using the unary encoding leads to an extremely large embedding formulation, but using a carefully selected binary encoding leads to a simple formulation with near-optimal size. In the following proposition we let $\text{size}_B\left(Q\left(\mathcal{P}, H\right)\right)$ denote the number of facets of $Q\left(\mathcal{P}, H\right)$ defined by variable bounds as we did for SOS2 constraints.

PROPOSITION 6. *Let $\mathcal{P} = \mathcal{P}(\mathcal{T})$ for a grid triangulation $\mathcal{T}$ on $[\![m+1]\!]^2$ and $n = 2m^2$ so that $|\mathcal{P}(\mathcal{T})| = n$. Then $\left(\sqrt{n/2} + 1\right)^2 \leq \text{mc}\left(\mathcal{P}\right)$ and if H is the unary encoding then*

$$\text{size}_B\left(Q\left(\mathcal{P}, H\right)\right) = \left(\sqrt{n/2} + 1\right)^2 \quad and \quad \text{size}\left(Q\left(\mathcal{P}, H\right)\right) = 4 + \left(\sqrt{n/2} + 1\right)^2 + \binom{2\sqrt{n/2}}{\sqrt{n/2}}.$$

In contrast, if m is a power of two and $\mathcal{T}$ is the union-jack triangulation, then there exist a binary encoding $H \in \mathcal{H}_{\log_2 n}(n)$ such that

$$\text{size}_B\left(Q\left(\mathcal{P}, H\right)\right) = \left(\sqrt{n/2} + 1\right)^2 \quad and \quad \text{size}\left(Q\left(\mathcal{P}, H\right)\right) = 4 + \left(\sqrt{n/2} + 1\right)^2 + 2\log_2(n/2). \quad (14)$$

We postpone a formal proof of Proposition 6 to Section 5.4 and instead illustrate it with the following example that shows how Lemma 2 can be used to recover an embedding formulation from the ideal formulation from Vielma and Nemhauser (2011). In particular, it shows how studying this formulation reveals the encoding needed to recover the embedding formulation.

EXAMPLE 1. The union-jack triangulation $\mathcal{T} = \left\{T_{u,v}^1, T_{u,v}^2\right\}_{u,v=1}^m$ for $m = 2$ depicted in Figure 3(b) can be described in a standard format used to construct the formulation in Vielma and Nemhauser (2011) by letting $T_{u,v}^1 = \{(2,2), (2, 2v-3), (2u-3, 2v-3)\}$ and $T_{u,v}^2 = \{(2,2), (2u-3, 2), (2u-3, 2v-3)\}$ for all $u, v \in [\![2]\!]$. In addition, the formulation from Vielma and Nemhauser (2011) for this triangulation is given by

$$\lambda_{(2,1)} + \lambda_{(2,3)} \leq 1 - y_1, \qquad \lambda_{(1,2)} + \lambda_{(3,2)} \leq y_1 \tag{15a}$$

$$\lambda_{(1,1)} + \lambda_{(2,1)} + \lambda_{(3,1)} \leq 1 - y_2, \quad \lambda_{(1,3)} + \lambda_{(2,3)} + \lambda_{(3,3)} \leq y_2 \tag{15b}$$

$$\lambda_{(1,1)} + \lambda_{(1,2)} + \lambda_{(1,3)} \leq 1 - y_3, \quad \lambda_{(3,1)} + \lambda_{(3,2)} + \lambda_{(3,3)} \leq y_3 \tag{15c}$$

$$y \in \{0,1\}^3, \qquad \sum_{u,v=1}^3 \lambda_{(u,v)} = 1, \quad \lambda_{(u,v)} \geq 0 \quad \forall u, v \in [\![3]\!]. \tag{15d}$$

If we let $\mathcal{P}(\mathcal{T}) := \left(P^i\right)_{i=1}^8$ as defined in Proposition 5, we can check that if (y, λ) is feasible for (15) and $y = (0,0,0)$ then $\lambda \in P^1 = P\left(T_{1,1}^1\right) = \left\{\lambda \in \Delta_2^3 : \lambda_{(u,v)} \leq 0 \quad \forall (u,v) \notin \{(2,2), (2,1), (1,1)\}\right\}$. Then letting $h^1 = (0,0,0)$ we obtain condition (13) of Lemma 2 for $i = 1$. Similarly, we may iterate over all values of $y \in \{0,1\}^3$ to obtain the complete $P\left(T_{u,v}^t\right)$-$h^{2(m(u-1)+(v-1))+t}$ or triangle-vector assignment for Lemma 2 depicted in Figure 3(b) and given by

$$h^1 = (0,0,0), \; T_{1,1}^1 = \{(2,2), (2,1), (1,1)\}; \qquad h^2 = (1,0,0), \; T_{1,1}^2 = \{(2,2), (1,2), (1,1)\};$$

$$h^3 = (0,0,1), \; T_{2,1}^1 = \{(2,2), (2,1), (3,1)\}; \qquad h^4 = (1,0,1), \; T_{2,1}^2 = \{(2,2), (3,2), (3,1)\};$$

$$h^5 = (0,1,0), \; T_{1,2}^1 = \{(2,2), (2,3), (1,3)\}; \qquad h^6 = (1,1,0), \; T_{1,2}^1 = \{(2,2), (1,2), (1,3)\};$$

$$h^7 = (0,1,1), \; T_{2,2}^1 = \{(2,2), (2,3), (3,3)\}; \qquad h^8 = (1,1,1), \; T_{2,2}^1 = \{(2,2), (3,2), (3,3)\}.$$

Because formulation (15) is ideal, by Lemma 2 we have that its LP relaxation is equal to $Q(\mathcal{P}(\mathcal{T}), H)$ for this H.

Proposition 6 shows that the specific encoding used can have a significant impact on the size of an embedding formulation for triangulations. As illustrated in Example 1, we can use an existing ideal formulation and Lemma 2 to recover a favorable encoding. Unfortunately, the formulation from Vielma and Nemhauser (2011) used to obtain the favorable encoding only works for the union-jack triangulation and it is sometimes preferable to use different triangulations such as the ones depicted in Figure 4 (e.g. Toriello and Vielma (2012)). In the following subsection we explore how adapting the favorable encoding for the union-jack triangulation to similar triangulations can sometimes help computationally construct a small embedding formulation.
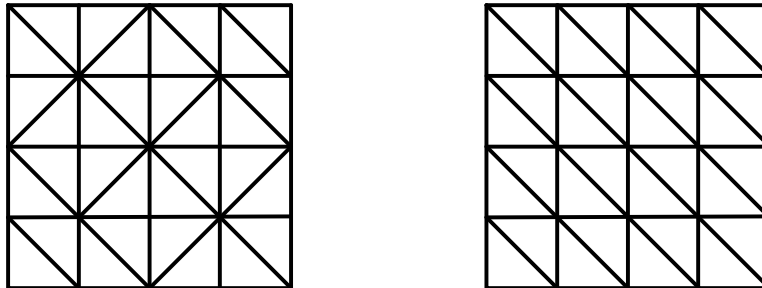
(a) Modified Union-Jack Triangulation for $m = 4$.   (b) K1 Triangulation for for $m = 4$.

Figure 4   Different Triangulations.

4.1. Constructing Embedding Formulations Computationally

One way to construct embedding formulations is to computationally construct the convex hull in (6) for a specific encoding. Picking a random encoding will likely result in an extremely large formulation (cf. Section 3.1). For this reason we now investigate the effectiveness of using a known favorable encoding for a similar constraint. For this we consider the modification of the union-jack triangulation illustrated in Figure 4(a). This triangulation is obtained by changing the way the bottom-left and top-right squares of the triangulation are divided into two triangles (in the original triangulation they are divided into top-left and bottom-right triangles, and in the modified triangulation they are divided into bottom-left and top-right triangles).

To construct an embedding formulation for this modified triangulation we adapt the encoding H associated to the logarithmic formulation for the union-jack triangulation (i.e. the encoding illustrated in Example 1). The adaptation uses the same triangle-vector assignment for all triangles

except for the ones belonging to the bottom-left and top-right squares. This adaptation is illustrated in Figure 5(a) for $m = 2$ where the top triangulation is the union-jack with the encoding described in Example 1 and the bottom triangulation is the modified union-jack with the adapted encoding. For both squares where the encoding is changed, the adaptation assigns to the bottom-left triangle the same h^i assigned to the top-left triangle in the original encoding for the union-jack triangulation. Similarly, the adaptation assigns to top-right triangle the same h^i assigned to the bottom-right triangle in the original encoding. To construct the formulation we simply compute the convex hull of $\bigcup_{=1}^n P^i \times \{h^i\}$ for this modified $H = (h^i)_{i=1}^n$ and $\mathcal{P}(\mathcal{T}) := (P^i)_{i=1}^n$ from Proposition 5 using the software cddlib (Fukuda 2005).

We tried this for $m \in \{4, 8, 16, 32\}$ and for all four cases the resulting embedding formulation only had four more inequalities than the formulation for the original union-jack triangulation. In addition, computing the convex hulls with cddlib for each $m \in \{4, 8, 16, 32\}$ took respectively less than a second, 10 seconds, 24 minutes and 3.5 days on an Intel i7-3770 3.40GHz workstation with 32GB of RAM. The computational time can grow quickly with m, but fortunately this computation only has to be done once and the formulation can then be stored. Similar to traditional MIP formulations, the resulting stored embedding formulation can be used for free in any problem that requires a piecewise linear functions of two variables based on the modified union-jack triangulation. More specifically, the same formulation can be used independently of the specific data (e.g. actual function values) associated to the piecewise linear functions (i.e. in the formulation from Proposition 5 we need to update (12a), but we do not need to change the embedding formulation for disjunction (12b)). Hence, 3.5 days does not seem that large when compared with the research time required to develop a small and ideal ad-hoc MIP formulation. Of course, this statement is conditional on the resulting formulation being small (so that it can be effectively stored and reused) and the formulation yielding a computational advantage (which is correlated with, but not guaranteed by a small size).

To check if the formulation obtained with cddlib preserves the computational advantage of the logarithmic formulation for the original union-jack triangulation, we replicate the computational experiments in Vielma and Nemhauser (2011) and Vielma et al. (2010) for the modified union-jack triangulation. These experiments consider a series of transportation problems whose objective functions are the sum of 25 piecewise linear functions of two variables on the $[\![m+1]\!]^2$ grid for $m \in \{4, 8, 16, 32\}$. For each m the experiment considers 100 randomly generated instances. With the exception of the logarithmic formulation, all formulations considered in the original experiment are applicable for the modified union-jack triangulation. So we test all these formulations with the logarithmic formulation replaced by the embedding formulation constructed computationally using cddlib. All formulations were implemented using the JuMP modeling language (Dunning

et al. 2015, Lubin and Dunning 2015, Dunning et al. 2017) and solved with Gurobi v6.5 (Gurobi Optimization 2015) on an Intel i7-3770 3.40GHz workstation with 32GB of RAM. Solve times for all combinations of formulations and solver are presented in Figure 5(b) for $m \in \{4, 8, 16, 32\}$. We refer the reader to Vielma et al. (2010) for details on the benchmark formulations, but we note that DCC and MC is obtained by variants of Theorem 1, CC is the generalization of formulation (1) and DCCLog is obtained by combining a variant of Theorem 1 with the same encoding used for the embedding formulation. We can see that the embedding formulation can provide a significant computational advantage.
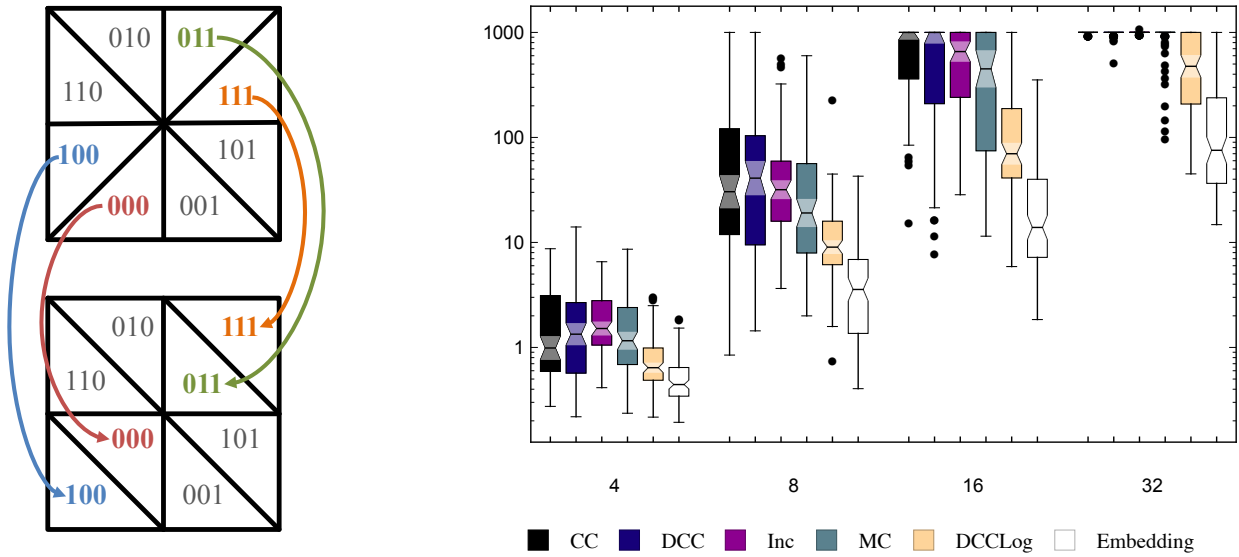
(a) Encoding Adaptation for $m = 2$.   (b) Solve Times for $m \in \{4, 8, 16, 32\}$ [s]

Figure 5   **Encoding and Solve Times for Modified Union-Jack Triangulation.**

5. Omitted Proofs and Additional Examples
5.1. Proof of Proposition 2

To prove Proposition 2 we use the following lemma, which gives a precise characterization of the facial structure of $Q(\mathcal{P}, H)$ for SOS2 constraints.

LEMMA 3. *Let* $\mathcal{P} := (P^i)_{i=1}^n$ *be the SOS2 constraint on* Δ^{n+1}, $H := (h^i)_{i=1}^n \in \mathcal{H}_k(n)$, $h^0 = h^1$, $h^{n+1} = h^n$, $c^i = h^{i+1} - h^i$ *for* $i \in [\![0, n]\!]$. *In addition, for any* $J^-, J^+ \subseteq [\![n+1]\!]$ *let*

$$E\left(J^-, J^+\right) := \bigcup_{j \in J^-} \left\{ \left(\mathbf{e}^j, h^{j-1}\right) \right\} \quad \cup \bigcup_{j \in J^+} \left\{ \left(\mathbf{e}^j, h^j\right) \right\}$$
$$= \bigcup_{j \in J^-} \left\{ \left(\mathbf{e}^j, h^j - c^{j-1}\right) \right\} \cup \bigcup_{j \in J^+} \left\{ \left(\mathbf{e}^j, h^j\right) \right\}$$

so that $Q(\mathcal{P}, H) = \text{conv}\left(E\left([\![n+1]\!], [\![n+1]\!]\right)\right)$ *and* $E\left([\![n+1]\!], [\![n+1]\!]\right) = \text{ext}\left(Q(\mathcal{P}, H)\right)$.

Then for any for any $J^-, J^+ \subseteq [\![n+1]\!]$ we have

$$\dim\left(E\left(J^-, J^+\right)\right) = \left|J^+ \cup J^-\right| - 1 + \dim\left(\left\{c^{j-1}\right\}_{j \in J^+ \cap J^-}\right).$$

Furthermore, $F \subseteq \mathbb{R}^{n+1+k}$ is a face of $Q\left(\mathcal{P}, H\right)$ if and only if there exist $J^-, J^+ \subseteq [\![n+1]\!]$ such that $F = F\left(J^-, J^+\right) := \operatorname{conv}\left(E\left(J^-, J^+\right)\right)$ and there exist $b \in \mathbb{R}^k$ such that

$$b \cdot c^{j-1} = 0 \quad \forall j \in J^+ \cap J^-, \quad b \cdot c^{j-1} < 0 \quad j \in J^- \setminus J^+ \quad \text{and} \quad b \cdot c^{j-1} > 0 \quad \forall j \in J^+ \setminus J^-. \tag{16}$$

Proof. For the dimension of $E\left(J^-, J^+\right)$, let $J \subseteq J^- \cap J^+$ be such that $\operatorname{span}\left(\left\{c^{j-1}\right\}_{j \in J}\right) = \operatorname{span}\left(\left\{c^{j-1}\right\}_{j \in J^- \cap J^+}\right)$ and $|J| = \dim\left(\left\{c^{j-1}\right\}_{j \in J^- \cap J^+}\right)$, and

$$E = \bigcup_{j \in J^- \setminus J^+}\left\{\left(\mathbf{e}^j, h^{j-1}\right)\right\} \cup \bigcup_{j \in J^+ \setminus J^-}\left\{\left(\mathbf{e}^j, h^j\right)\right\} \cup \bigcup_{j \in J^- \cap J^+}\left\{\left(\mathbf{e}^j, h^j\right)\right\} \cup \bigcup_{j \in J}\left\{\left(\mathbf{e}^j, h^{j-1}\right)\right\}.$$

We can check that E is a set of $\left|J^+ \cup J^-\right| + \dim\left(\left\{c^{j-1}\right\}_{j \in J^+ \cap J^-}\right)$ affinely independent vectors. Finally, for any $(\lambda, y) \in E\left(J^-, J^+\right) \setminus E$ there exist $j \in J^+ \cap J^- \setminus J$ such that $(\lambda, y) = \left(\mathbf{e}^j, h^{j-1}\right) = \left(\mathbf{e}^j, h^j - c^{j-1}\right)$. Let $\mu \in \mathbb{R}^J$ be such that $c^{j-1} = \sum_{i \in J} \mu_i c^{i-1}$. Then $\left(\mathbf{e}^j, h^j - c^{j-1}\right) = \mu_0\left(\mathbf{e}^j, h^j\right) + \sum_{i \in J}\left(\mu_i\left(\mathbf{e}^i, h^i - c^{i-1}\right) - \mu_i\left(\mathbf{e}^i, h^i\right)\right)$ for $\mu_0 = 1$ and result follows.

For the facial characterization we have that $F \subseteq \mathbb{R}^{n+1+k}$ is a face of $Q\left(\mathcal{P}, H\right)$ if and only if there exist $(a, b) \in \mathbb{R}^{n+1+k}$ such that

$$\begin{aligned} F &= \arg\max\left\{a \cdot \lambda + b \cdot y : (\lambda, y) \in Q\left(\mathcal{P}, H\right)\right\} \\ &= \operatorname{conv}\left(\arg\max\left\{a \cdot \lambda + b \cdot y : (\lambda, y) \in E\left([\![n+1]\!], [\![n+1]\!]\right)\right\}\right). \end{aligned} \tag{17}$$

Let $c = \max\left\{a \cdot \lambda + b \cdot y : (\lambda, y) \in Q\left(\mathcal{P}, H\right)\right\}$, $J^- = \left\{j \in [\![n+1]\!] : a \cdot \mathbf{e}^j + b \cdot h^{j-1} = c\right\} = \left\{j \in [\![n+1]\!] : a \cdot \mathbf{e}^j + b \cdot h^j - b \cdot c^{j-1} = c\right\}$ and $J^+ = \left\{j \in [\![n+1]\!] : a \cdot \mathbf{e}^j + b \cdot h^j = c\right\}$. Then $F = \operatorname{conv}\left(E\left(J^-, J^+\right)\right)$, $b \cdot c^{j-1} = 0$ for all $j \in J^+ \cap J^-$, $b \cdot c^{j-1} < 0$ for all $j \in J^- \setminus J^+$ and $b \cdot c^{j-1} > 0$ for all $j \in J^+ \setminus J^-$. Conversely if $F = \operatorname{conv}\left(E\left(J^-, J^+\right)\right)$ for $J^-, J^+ \subseteq [\![n+1]\!]$ and $b \in \mathbb{R}^k$ satisfies (16) let $a_j = b \cdot h^j$ for $j \in J^+ = \left(J^+ \cap J^-\right) \cup \left(J^+ \setminus J^-\right)$, $a_j = b \cdot h^{j-1}$ for $j \in J^- \setminus J^+$ and $a_j = \min\left(b \cdot h^j, b \cdot h^{j-1}\right) - 1$ for $j \in [\![n+1]\!] \setminus \left(J^- \cup J^+\right)$. The a, b and F satisfy (17). $\quad\square$

Proof of Proposition 2. Throughout the proof we let $h^0 = h^1$, $h^{n+1} = h^n$ as in the statement of Lemma 3 so we can define $c^i = h^{i+1} - h^i$ for all $i \in [\![0, n]\!]$ and it coincides with the proposition's statement for $i \in [\![n-1]\!]$ and $c^0 = c^n = \mathbf{0}$. We will also use the straightforward fact that $L(H) = \operatorname{span}\left(\left\{c^i\right\}_{i \in [\![n-1]\!]}\right) = \operatorname{span}\left(\left\{c^i\right\}_{j \in [\![0, n]\!]}\right)$.

Any $(\lambda, y) \in Q\left(\mathcal{P}, H\right)$ satisfies the equations in (8a). Furthermore, because $L(H) = \operatorname{span}\left(\left\{c^i\right\}_{i \in [\![n-1]\!]}\right)$ we have that the dimension of the affine subspace described by (8a) is $n + \dim\left(\left\{c^i\right\}_{i \in [\![n-1]\!]}\right)$. Finally, by Lemma 3 $\dim\left(Q\left(\mathcal{P}, H\right)\right) = n + \dim\left(\left\{c^i\right\}_{i \in [\![n-1]\!]}\right)$, which shows the statements about $\dim\left(Q\left(\mathcal{P}, H\right)\right)$ and $\operatorname{aff}\left(Q\left(\mathcal{P}, H\right)\right)$.

Using Lemma 3 we have that a face $F(J^-, J^+) = \text{conv}(E(J^-, J^+))$ for $J^-, J^+ \subseteq [\![n+1]\!]$ may be a facet of $Q(\mathcal{P}, H)$ only if $|J^+ \cup J^-| = n+1$ and $\dim\left(\{c^{j-1}\}_{j \in J^+ \cap J^-}\right) = \dim(L(H)) - 1$ or $|J^+ \cup J^-| = n$ and $\dim\left(\{c^{j-1}\}_{j \in J^+ \cap J^-}\right) = \dim(L(H))$.

In the first option $J^+ \cup J^- = [\![n+1]\!]$ and $J^+ \cap J^- \subsetneq [\![n+1]\!]$ because $\dim\left(\{c^{j-1}\}_{j \in [\![n+1]\!]}\right) = \dim(L(H))$. Then, $(J^- \setminus J^+) \cup (J^+ \setminus J^-) \neq \emptyset$ and condition (16) of Lemma 3 holds for $b \in \mathbb{R}^k \setminus \{\mathbf{0}\}$. Furthermore, because of the first part of condition (16) and $\dim\left(\{c^{j-1}\}_{j \in J^+ \cap J^-}\right) = \dim(L(H)) - 1$ we further have that $b = sb^l$ for some $l \in [\![L]\!]$ and $s \in \{-1, 1\}$, and $J^+ \cap J^- = \left\{j \in [\![n+1]\!] : c^{j-1} \in M(b^l)\right\}$ (note that because $c^0 = c^n = 0$ we always have $1, n+1 \in J^+ \cap J^-$). If $s = 1$ the second inequality in (8b) for b^l is satisfied at equality by all points in $E(J^-, J^+)$ and strictly by all points in $E([\![n+1]\!], [\![n+1]\!]) \setminus E(J^-, J^+)$, and hence defines $F(J^-, J^+)$. Similarly, if $s = -1$ the first inequality in (8b) for b^l defines $F(J^-, J^+)$.

In the second option we have that span $\left(\{c^{j-1}\}_{j \in J^+ \cap J^-}\right) = L(H)$ and hence the first part of condition (16) of Lemma 3 implies $b \in L(H)^\perp$. However, span $\left(\{c^{j-1}\}_{j \in [\![n+1]\!]}\right) = L(H)$ so the second part of condition (16) implies $J^- \setminus J^+ = J^+ \setminus J^- = \emptyset$ and hence $J^- = J^+ = [\![n+1]\!] \setminus \{j_0\}$ for some $j_0 \in J$. Noting that $c^0 = c^n = 0$ we have that condition span $\left(\{c^{j-1}\}_{j \in J^+ \cap J^-}\right) = \text{span}\left(\{c^{j-1}\}_{j \in [\![n+1]\!] \setminus \{j_0\}}\right) = L(H)$ holds if and only if $j_0 \in J$ for J defined in the proposition statement. Furthermore, in such case the inequality in (8c) corresponding to $j = j_0$ defines $F(J^-, J^+)$.

Hence the equations of (8) are precisely those defining aff$(Q(\mathcal{P}, H))$, every facet of $Q(\mathcal{P}, H)$ is defined by an inequality of (8) and every inequality of (8) is facet defining for $Q(\mathcal{P}, H)$. Finally, the last statement follows because the two options considered for facets of $Q(\mathcal{P}, H)$ yield two distinct classes of facets. $\quad\square$

5.2. An Embedding Formulation for SOS2 Constraints Whose Validity is Not Evident

EXAMPLE 2. Let $\mathcal{P} := (P^i)_{i=1}^9$ be the SOS2 constraint on Δ^{10} and $H = \big((0,1,1,1)^T,$ $(0,1,0,0)^T, (0,0,0,0)^T, (0,1,0,1)^T, (0,0,0,1)^T, (1,0,0,0)^T, (1,1,0,1)^T, (1,0,1,1)^T, (1,1,1,1)^T\big)$. Then $(c^i)_{i=1}^8 = \big\{(0,0,-1,-1)^T, (0,-1,0,0)^T, (0,1,0,1)^T, (0,-1,0,0)^T, (1,0,0,-1)^T, (0,1,0,1)^T,$ $(0,-1,1,0)^T, (0,1,0,0)^T\big\}$ and the set of hyperplanes spanned by them is given by $\{M(b^l)\}_{l=1}^5$ for $b^1 = (1,0,0,-1,1)^T$, $b^2 = (1,0,0,1)^T$, $b^3 = (1,-1,-1,1)^T$, $b^4 = (1,0,0,0)^T$ and $b^5 = (0,0,1,0)^T$. Finally, aff$(H) = L(H) = \mathbb{R}^4$ and $\mathbb{R}^4 = \text{span}\left(\{c^i\}_{i \in [\![9]\!] \setminus \{j-1\}}\right)$ if and only if $j \in [\![2, 10]\!] \setminus \{6\}$. Then (8) in this case is given by

$$\sum\nolimits_{j=1}^{10} \lambda_j = 1, \tag{18a}$$

$$\lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} \leq y_1 - y_3 + y_4 \tag{18b}$$

$$\lambda_4 + \lambda_5 + \lambda_6 + 2\lambda_7 + 2\lambda_8 + \lambda_9 + \lambda_{10} \geq y_1 - y_3 + y_4 \tag{18c}$$

$$\lambda_1 + \lambda_5 + \lambda_6 + \lambda_7 + 2\lambda_8 + 2\lambda_9 + 2\lambda_{10} \leq y_1 + y_4 \tag{18d}$$

$$\lambda_1 + \lambda_2 + \lambda_4 + \lambda_5 + \lambda_6 + 2\lambda_7 + 2\lambda_8 + 2\lambda_9 + 2\lambda_{10} \geq y_1 + y_4 \tag{18e}$$

$$-\lambda_1 - \lambda_2 - \lambda_3 + \lambda_6 + \lambda_7 + \lambda_8 \leq y_1 - y_2 - y_3 + y_4 \tag{18f}$$

$$-\lambda_1 - \lambda_2 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 \geq y_1 - y_2 - y_3 + y_4 \tag{18g}$$

$$\lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} \leq y_1 \tag{18h}$$

$$\lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} \geq y_1 \tag{18i}$$

$$\lambda_1 + \lambda_9 + \lambda_{10} \leq y_3 \tag{18j}$$

$$\lambda_1 + \lambda_2 + \lambda_8 + \lambda_9 + \lambda_{10} \geq y_3 \tag{18k}$$

$$\lambda_j \geq 0 \qquad\qquad \forall j \in [\![10]\!] \setminus \{6\}, \tag{18l}$$

If $y = h^4 = (0,1,0,1)^T$ then (18) should enforce $\lambda_i = 0$ for all $i \notin \{4,5\}$. Inequality (18f) enforces that $\lambda_6 = 1$ cannot hold, but it does not force $\lambda_6 = 0$ as $\lambda_1 = \lambda_6 = 1/2$ is valid for this inequality. However, this last point is infeasible for (18c). These two inequalities plus (18a) do indeed imply $\lambda_6 \leq 0$ (and hence $\lambda_6 = 0$ because of the lower bounds) when $y = h^4$ as adding (18a), (18c) and (18f) yields

$$\lambda_6 \leq 1 - y_2.$$

Furthermore, removing any one of these constraints allows $\lambda_6 > 0$ when $y = h^4$.

5.3. Proof of Proposition 4

To prove Proposition 4 we need the following definition.

DEFINITION 7. Let $n = 2^k$ for some $k \in \mathbb{Z}$. We say $H = \{h^i\}_{i=1}^n \in \mathcal{H}_k(n)$ is a *anti-gray code*[||] if and only if $\sum_{j=1}^k \left| h_j^{2i-1} - h_j^{2i} \right| = n$ for all $i \in [\![n/2]\!]$ and $\sum_{j=1}^k \left| h_j^{2i} - h_j^{2i+1} \right| = n - 1$ for all $i \in [\![n/2 - 1]\!]$.

Anti-gray codes exist for all k and can easily be constructed from gray codes (e.g. Robinson and Cohn (1981)).

Proof of Proposition 4. Let H be an anti-gray code and $c^i = h^{i+1} - h^i$ for $i \in [\![n-1]\!]$. Because H is an anti-gray code there exist $I \subseteq [\![n-1]\!]$ with $|I| = 2^{k-1}$ such that $c^i \in \{-1,1\}^k$ for all $i \in I$. In addition, because $h^i \neq h^j$ for $i \neq j$ we have that $c^i \neq -c^j$ for all $i,j \in I$. Hence for all $s \in \{-1,1\}^k$ there exist $i \in I$ such that $s = c^i$ or $s = -c^i$. The result then follows from Proposition 2 by noting that $\{c^i\}_{i \in I}$ and $\{\pm c^i\}_{i \in I}$ span the same set of linear hyperplanes and that the number of linear hyperplanes spanned by $\{-1,1\}^k$ is equal to the number of affine hyperplanes spanned by $\{0,1\}^{k-1}$ (e.g. da Silva (2005)) □

[||]The class of codes obtained by switching n and $n-1$ in this definition is sometimes also referred to as anti-gray code.

5.4. Proof of Proposition 6

Proof Proposition 6. For any $H \in \mathcal{H}_k(n)$ we have $Q(\mathcal{P}, H) \subseteq \Delta_2^{m+1} :=$ $\left\{\lambda \in \mathbb{R}_+^{[\![m+1]\!]^2} : \sum_{u,v=1}^{m+1} \lambda_{(u,v)} = 1\right\}$. We begin by showing that for any $H \in \mathcal{H}_k(n)$ all inequalities $\lambda_{(u,v)} \geq 0$ of Δ_2^{m+1} are facet defining for $Q(\mathcal{P}, H)$. For that let $n = 2m^2$ and $(S_i)_{i=1}^n$ be such that $S_{2(m(u-1)+(v-1))+t} = T_{u,v}^t$ for all $u, v \in [\![m]\!]$ and $t \in \{1, 2\}$ so that $Q(\mathcal{P}, H) = \text{conv}\left(\bigcup_{i=1}^n P(S_i) \times \{h^i\}\right)$. Then $\lambda_{(u,v)} \geq 0$ describes a face of $Q(\mathcal{P}, H)$ because it is valid and is satisfied at equality for all points $(\lambda, y) = (\mathbf{e}^{\bar{u},\bar{v}}, h^i)$ for $(\bar{u}, \bar{v}) \in S_i \setminus \{(u, v)\}$. Let

$$a \cdot \lambda + b \cdot y \leq c \tag{19}$$

be a valid inequality of $Q(\mathcal{P}, H)$ that induces a facet containing the face induced by $\lambda_{(u,v)} \geq 0$. Because $Q(\mathcal{P}, H) \subseteq \Delta_2^{m+1} \cap \text{aff}(H)$, without loss of generality we may assume $c = 0$ and $b \in L(H)$ by possibly adding multiples of $\sum_{u,v=1}^{m+1} \lambda_{(u,v)} = 1$ and the equations defining aff (H). For any $i, j \in [\![n]\!]$, $(\underline{u}, \underline{v}) \in S_i \setminus \{(u, v)\}$ and $(\bar{u}, \bar{v}) \in S_j \setminus \{(u, v)\}$ there exist $(i_l)_{l=1}^r \subseteq [\![n]\!]$ and $((u_l, v_l))_{l=1}^{r-1} \subseteq [\![n+1]\!]^2 \setminus \{(u, v)\}$ such that $i_1 = i$, $i_r = j$, $(\underline{u}, \underline{v}) = (u_{i_1}, v_{i_1})$, $(\bar{u}, \bar{v}) = (u_{i_r}, v_{i_r})$ and $(u_l, v_l) \in S_{i_l} \cap S_{i_{l+1}}$ for all $l \in [\![r-1]\!]$. Because $(\lambda, y) = (\mathbf{e}^{(u_l, v_l)}, h^l)$ and $(\lambda, y) = (\mathbf{e}^{(u_l, v_l)}, h^{l+1})$ satisfy (19) at equality for all $l \in [\![r-1]\!]$, we have $b \cdot h^i = b \cdot h^j$ and $a_{(\underline{u},\underline{v})} = a_{(\bar{u},\bar{v})} = b \cdot h^i$. Because the first identity holds for all $i, j \in [\![n]\!]$ we have $b \cdot (h^i - h^1) = 0$ for all $i \in [\![n]\!]$, which together with $b \in L(H)$ implies $b = \mathbf{0}$. Similarly, combining $b = \mathbf{0}$ and the fact that the second identity holds for all $(\underline{u}, \underline{v}), (\bar{u}, \bar{v}) \neq (u, v)$ we obtain $a_{(\bar{u},\bar{v})} = 0$ for all $(\bar{u}, \bar{v}) \neq (u, v)$. Finally, validity of (19) implies $a_{(u,v)} \leq 0$ and being facet defining further implies $a_{(u,v)} < 0$. Then (19) is a positive multiple of $\lambda_{(u,v)} \geq 0$ and hence $\lambda_{(u,v)} \geq 0$ is facet defining.

The result on $\lambda_{(u,v)} \geq 0$ shows that $\left(\sqrt{n/2} + 1\right)^2 \leq \text{size}_B(Q(\mathcal{P}, H)) \leq \text{mc}(\mathcal{P})$ for any $H \in \mathcal{H}_k(n)$.

The sizes for the unary encoded formulation comes from Proposition 10 in Lee and Wilson (2001), the comments before its statement and Lemma 2. The existence and sizes for the binary encoded formulation come from the proof of Theorem 1 in Vielma and Nemhauser (2011), Lemma 2, the lower bound on $\text{size}_B(Q(\mathcal{P}, H))$, noting that the bounds on the binary variables of the formulation from Vielma and Nemhauser (2011) are redundant (cf. the logarithmic formulation for SOS2) and that none of the non-bound inequalities of this formulation are redundant. $\square$

References

Aichholzer, O., F. Aurenhammer. 1996. Classifying hyperplanes in hypercubes. *SIAM Journal on Discrete Mathematics* **9** 225–232.

Balas, E. 1985. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal on Algebraic and Discrete Methods* **6** 466–486.

Beale, E. M. L., J. A. Tomlin. 1970. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. J. Lawrence, ed., *OR 69: Proceedings of the fifth international conference on operational research*. Tavistock Publications, 447–454.

Conforti, M., G. Cornuejols, G. Zambelli. 2014. *Integer Programming*. Graduate Texts in Mathematics, Springer International Publishing.

Croxton, K. L., B. Gendron, T. L. Magnanti. 2003. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science* **49**(9) 1268–1273.

da Silva, I. P.F. 2005. Recursivity and geometry of the hypercube. *Linear algebra and its applications* **397** 223–233.

D'Andreagiovanni, F., C. Mannino, A. Sassano. 2013. GUB covers and power-indexed formulations for wireless network design. *Management Science* **59**(1) 142–156.

Dunning, I., Joey H., M. Lubin. 2017. JuMP: A modeling language for mathematical optimization. *SIAM Review* **59** 295–320.

Dunning, I., J. Huchette, M. Lubin. 2015. JuMP Modeling language for Mathematical Programming. https://github.com/JuliaOpt/JuMP.jl.

Fukuda, K. 2005. cddlib reference manual. https://www.inf.ethz.ch/personal/fukudak/cdd_home/.

Gurobi Optimization. 2015. The Gurobi Optimizer. http://www.gurobi.com.

Huber, B., J. Rambau, F. Santos. 2000. The cayley trick, lifting subdivisions and the bohne-dress theorem on zonotopal tilings. *Journal of the European Mathematical Society* **2** 179–198.

Huchette, J., S. S. Dey, J. P. Vielma. 2016. Strong mixed-integer formulations for the floor layout problem. *arXiv preprint arXiv:1602.07760* .

Jeroslow, R. G., J. K. Lowe. 1984. Modeling with integer variables. *Mathematical Programming Studies* **22** 167–184.

Jünger, M., T. Liebling, D. Naddef, G. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi, L. Wolsey. 2010. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer-Verlag, New York.

Kaibel, V., S. Weltge. 2015. Lower bounds on the sizes of integer programs without additional variables. *Math. Program. Ser. B* **154** 407–425.

Karavelas, M. I., C. Konaxis, E. Tzanaki. 2013. The maximum number of faces of the minkowski sum of three convex polytopes. G. D. da Fonseca, T. Lewiner, L. M. Peñaranda, T. M. Chan, R. Klein, eds., *Symposuim on Computational Geometry 2013, SoCG '13, Rio de Janeiro, Brazil, June 17-20, 2013*. ACM, 187–196.

Lee, J., D. Wilson. 2001. Polyhedral methods for piecewise-linear functions I: the lambda method. *Discrete Applied Mathematics* **108** 269–285.

Lubin, M., I. Dunning. 2015. Computing in operations research using julia. *INFORMS Journal on Computing* **27**(2) 238–248.

Manne, A. S. 1960. On the job-shop scheduling problem. *Operations Research* **8**(2) 219–223.

Muldoon, F. 2012. Polyhedral approximations of quadratic semi-assignment problems, disjunctive programs, and base-2 expansions of integer variables. Ph.D. thesis, Clemson University.

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

Pinedo, M.L. 2012. *Scheduling: Theory, Algorithms, and Systems*. SpringerLink : Bücher, Springer New York.

Roberti, R., E. Bartolini, A. Mingozzi. 2014. The fixed charge transportation problem: An exact algorithm based on a new integer programming formulation. *Management Science* **61**(6) 1275–1291.

Robinson, J. P., M. Cohn. 1981. Counting sequences. *IEEE Transactions on Computers* **100** 17–23.

Todd, M. J. 1977. Union jack triangulations. S. Karamardian, ed., *Fixed Points: algorithms and applications*. Academic Press, 315–336.

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

Vielma, J. P. 2015. Mixed integer linear programming formulation techniques. *SIAM Review* **57** 3–57.

Vielma, J. P. 2017. Small and strong formulations for unions of convex sets from the cayley embedding. *arXiv preprint arXiv:1704.03954* .

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

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

Voigt, T., G. M. Ziegler. 2006. Singular 0/1-matrices, and the hyperplanes spanned by random 0/1-vectors. *Combinatorics, Probability and Computing* **15** 463–471.

Weibel, C. 2007. Minkowski sums of polytopes: Combinatorics and computation. Ph.D. thesis, École Polytechnique Fédérale de Lausanne.

Weltge, S. 2015. Sizes of linear descriptions in combinatorial optimization. Ph.D. thesis, Magdeburg, Universität, Diss.