Advanced mixed-integer programming formulations: Methodology, computation, and application

by

Joseph Andrew Huchette

B.A., Rice University (2013)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author ....
Sloan School of Management
May 15, 2018

Certified by ....
Juan Pablo Vielma
Richard S. Leghorn (1939) Career Development Associate Professor of
Operations Research
Thesis Supervisor

Accepted by ....
Dimitris Bertsimas
Boeing Professor of Operations Research
Codirector, Operations Research Center

Advanced mixed-integer programming formulations: Methodology, computation, and application

by

Joseph Andrew Huchette

Submitted to the Sloan School of Management
on May 15, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis introduces systematic ways to use mixed-integer programming (MIP) to solve difficult nonconvex optimization problems arising in application areas as varied as operations, robotics, power systems, and machine learning. Our goal is to produce MIP formulations that perform extremely well in practice, requiring us to balance qualities often in opposition: formulation size, strength, and branching behavior.

We start by studying a combinatorial framework for building MIP formulations, and present a complete graphical characterization of its expressive power. Our approach allows us to produce strong and small formulations for a variety of structures, including piecewise linear functions, relaxations for multilinear functions, and obstacle avoidance constraints.

Second, we present a geometric way to construct MIP formulations, and use it to investigate the potential advantages of general integer (as opposed to binary) MIP formulations. We are able to apply our geometric construction method to piecewise linear functions and annulus constraints, producing small, strong general integer MIP formulations that induce favorable behavior in a branch-and-bound algorithm.

Third, we perform an in-depth computational study of MIP formulations for nonconvex piecewise linear functions, showing that the new formulations devised in this thesis outperform existing approaches, often substantially (e.g. solving to optimality in orders of magnitude less time). We also highlight how high-level, easy-to-use computational tools, built on top of the JuMP modeling language, can help make these advanced formulations accessible to practitioners and researchers. Furthermore, we study high-dimensional piecewise linear functions arising in the context of deep learning, and develop a new strong formulation and valid inequalities for this structure.

We close the thesis by answering a speculative question: Given a disjunctive constraint, what can we reasonably sacrifice in order to construct MIP formulations with very few integer variables? We show that, if we allow our formulations to introduce spurious "integer holes" in their interior, we can produce strong formulations for any disjunctive constraint with only two integer variables and a linear number of inequalities (and reduce this further to a constant number for specific structures).

We provide a framework to encompass these *MIP-with-holes formulations*, and show how to modify standard MIP algorithmic tools such as branch-and-bound and cutting planes to handle the holes.

Thesis Supervisor: Juan Pablo Vielma
Title: Richard S. Leghorn (1939) Career Development Associate Professor of Operations Research

Acknowledgments

This thesis would certainly not exist without the guidance of my advisor, Juan Pablo Vielma. His energy and excitement are an inspiration, and he has always guided me towards interesting problems, while allowing me the freedom to explore them once I get there. He has been incredibly generous with his time and financial support, and the extent to which I have been able to travel, meet interesting people, and present my work to others has undoubtedly made me a much better researcher and academic. I thank him for serving as an exemplar of the advisor I hope to be in the coming years.

I would also like to thank Jim Orlin and Michel Goemans for serving on my thesis committee. In particular, their insightful comments and suggestions on early presentations of this work helped me immeasurably in communicating its main ideas as effectively as possible.

The Operations Research Center is a special place, entirely because of the people found behind its doors. Dimitris and Patrick deserve commendation for their leadership and vision, as do Laura and Andrew for ensuring that everything runs as smoothly as it does. My fellow students are brilliant, but also fun, interesting, and grounded, and I am grateful to have made so many great friends during my time here.

I would like to thank my coauthors during my time at MIT–especially Santanu Dey and Cosmin Petra–for their guidance and mentorship early during my PhD. I am also indebted to my undergraduate professors at Rice for seeding my interest in mathematics and operations research, and especially Beatrice Riviere and Hadley Wickham for their mentorship as I took my first steps as a researcher. I owe a special thanks to Miles and Iain for starting JuMP and recruiting me to join before any of us really knew where it was headed. It's been fantastic to have something "fun" to work on besides my research, and that it has blossomed into something others find useful makes it all the more rewarding.

I would like to thank all my friends–from all stages of life, in Cambridge and elsewhere–for making life interesting. You all have make the past five years as enjoy-able as they have been.

I am especially grateful to have had Elisabeth in my life these past two years: as my travel companion, research commiserator, and partner.

Finally, without the support of my parents and family–for as long as I can remember, and before that too, I'm sure–I certainly wouldn't be the person I am today.

Contents

1 Preliminaries. **21**

1.1 Mixed-integer programming and formulations 21

1.2 Disjunctive constraints 23

1.2.1 Combinatorial disjunctive constraints 24

1.2.2 Combinatorial disjunctive constraints and data independence . 26

1.3 Motivating problems 27

1.3.1 Modeling discrete alternatives 27

1.3.2 Piecewise linear functions 28

1.3.3 The SOSk constraint 34

1.3.4 Cardinality constraints 34

1.3.5 Discretizations of multilinear functions 34

1.3.6 Obstacle avoidance constraints 36

1.3.7 Annulus constraints 37

1.3.8 Optimizing over trained neural networks 38

1.4 Assessing the quality of MIP formulations 39

1.4.1 Strength 39

1.4.2 Size 41

1.4.3 Branching behavior 41

1.5 Existing approaches 44

1.6 Contributions of this thesis 46

1.7 Notation 48

2 A combinatorial way to construct formulations. 51

2.1 Independent branching schemes 52

2.2 What does *independent branching* mean? 53

2.2.1 Constraint branching via independent branching schemes . . . 53

2.2.2 Independence in formulation-induced branching schemes . . . 55

2.3 Independent branching scheme representability 58

2.3.1 A graphical characterization 58

2.3.2 Cardinality constraints 60

2.3.3 Polygonal partitions of the plane 60

2.3.4 Pairwise IB-representable constraints 63

2.4 Pairwise independent branching schemes 66

2.4.1 Graphical representations of pairwise IB-representable CDCs . 66

2.4.2 Representation at a given depth 67

2.5 A simple IB scheme and its limitations 71

2.6 Systematic construction of biclique covers 74

2.7 Biclique covers for graph products and discretizations of multilinear functions 77

2.8 Completing biclique covers via graph unions 79

2.8.1 Grid triangulations of the plane 81

2.8.2 Higher-dimensional grid triangulations 84

2.8.3 An even smaller formulation for bivariate grid triangulations . 85

2.8.4 The SOSk constraint 88

3 Building formulations geometrically: Embeddings. 95

3.1 The embedding approach 95

3.2 How many integer variables do we need? 97

3.2.1 A geometric characterization of when embeddings yield valid formulations 104

3.2.2 Negative results for combinatorial disjunctive constraints . . . 105

3.3 A geometric construction for ideal formulations of any combinatorial disjunctive constraint 107
3.4 Novel MIP formulations for univariate piecewise linear functions . . . 113
3.5 Small MIP formulations for the annulus 117

4 MIP formulations for nonconvex piecewise linear functions. 123
4.1 Univariate piecewise linear functions 123
4.1.1 Univariate computational experiments 124
4.1.2 Branching behavior of MIP formulations 128
4.2 New hybrid formulations for bivariate piecewise linear functions . . . 135
4.2.1 The combination of ideal formulations 135
4.2.2 Bivariate computational experiments 138
4.2.3 Optimal independent branching schemes 140
4.3 Computational tools for piecewise linear modeling: `PiecewiseLinearOpt` 142
4.4 Preliminary extensions: MIP formulations for neural networks 146
4.4.1 Existing formulations 146
4.4.2 An ideal formulation for a single ReLu 147
4.4.3 A separation procedure 152
4.4.4 Valid inequalities for multiple layers 154

5 Very small formulations and the MIP-with-holes approach. 157
5.1 Branching schemes and MIP-with-holes formulations 160
5.2 Choices of encodings 163
5.2.1 The reflected binary Gray and zig-zag encodings 163
5.2.2 The moment curve encoding 163
5.2.3 A more exotic encoding 164
5.3 Very small MIP-with-holes formulations 167
5.3.1 A big-M MIP-with-holes formulation for any disjunctive set . 169
5.3.2 A very small formulation for the SOS1 constraint 170
5.3.3 Very small formulations for general combinatorial disjunctive constraints 171

5.3.4 A very small formulation for the SOS2 constraint 174
5.3.5 A very small formulation for the annulus 179

A The logarithmic formulation of Misener et al. [106] is not ideal 181

B Constructing a pairwise IB-representable partition of the plane 183

C 8-segment piecewise linear function formulation branching 187

D Additional computational results with Gurobi 191

List of Figures

1-1 **(Left)** A univariate piecewise linear function, and **(Right)** a bivariate piecewise linear function with a grid triangulated domain. 28

1-2 Three grid triangulations of $\Omega = [1,3] \times [1,3]$: **(Left)** the Union Jack (J1) [127], **(Center)** the K1 [84], and **(Right)** a more idiosyncratic triangulation. 32

1-3 Two bivariate functions over $D = [0,1]^2$ that match on the gridpoints, but differ on the interior of D. 33

1-4 A trajectory for a UAV which avoids an obstacle in the center of the domain. 36

1-5 **(Left)** The annulus $\mathcal{A}$ and **(Right)** its corresponding quadrilateral relaxation $\hat{\mathcal{A}}$ given by (1.14) with $d = 8$. 37

1-6 The relaxations (gray region) for two different formulations of a non-convex set (solid lines), corresponding a univariate piecewise linear function. **(Left)** One is not sharp, while **(Right)** the second is sharp. If we solve the relaxted optimization problem $\min_{(x,y)\in R} y$ for each relaxation R, we get different optimal solution values (optimal solutions circled), and therefore different dual bounds for the MIP problem. . . 40

1-7 **(Top)** Good branching for one formulation of a univariate piecewise linear function, and **(Bottom)** bad, unbalanced branching from another formulation. 42

2-1 The branch-and-bound trees for **(Left)** (2.4) and **(Right)** (2.5), when **(Top row)** z_1 is first to branch on, and then z_2, and when **(Bottom row)** z_2 is first to branch on, and then z_1. Inside each node is the set $I \subset [\![5]\!]$ of all components v for which the algorithm has been able prove that $\lambda_v = 0$ at this point in the algorithm via branching decisions. The text on the lines show the current branching decision (e.g. $z_2 \geqslant 1$), and the set of components $v \in [\![5]\!]$ for which the (a) subproblem is able to prove that $\lambda_v = 0$ *independently* of any other branching decisions (e.g. $z_2 \geqslant 1$ is the only additional branching constraint added to the original relaxation). This figure is adapted from [135, Figure 2]. . . . 57

2-2 Partitions of a nonconvex region in the plane obtained by removing a central non-convex portion from a convex polyhedron. **(Left)** The first partition does not satisfy the internal vertices condition (2.7), **(Center)** the second partition admits a pairwise IB scheme, and **(Right)** the third partition admits a 3-way IB scheme but not a pairwise one. 63

2-3 Visualizations of the biclique covers presented in the text for **(Left)** SOS3(6) and **(Right)** SOS3(10). Each row corresponds to some level j, and the elements of A^j and B^j are the blue squares and green diamonds, respectively. 73

2-4 The recursive construction for biclique covers for SOS2. The first row is a single biclique that covers the conflict graph for SOS2(3) ($A^{1,1}$ in blue, $B^{1,1}$ in green). The second row shows the construction which duplicates the ground set $\{1, 2, 3\}$ and inverts the ordering on the second copy. The third row shows the identification of the nodes that yields a valid biclique for SOS2(5). This biclique is then combined with a second that covers all edges between nodes identified with the first copy and those identified with the second, giving a biclique cover for SOS2(5) with two levels. 75

2-5 Independent branching schemes for the three triangulations presented in Figure 1-2, each given its own row.The sets A^j and B^j are given by the blue squares and green diamonds, respectively, in the i-th subfigure of the corresponding row. 80

2-6 Two bicliques constructed via Corollary 5 for a grid triangulation with $d_1 = d_2 = 8$. On the left the construction follows by taking $u = (1, 1)$; on the right, with $u = (2, 3)$. For each level, the sets $A(u)$ and $B(u)$ are given by the blue squares and green diamonds, respectively. . . . 83

2-7 The aggregated SOS2 independent branching formulation for subrectangle selection. The sets $A^{1,k}$ (resp. $B^{1,k}$) are the squares (resp. diamonds) in the first row; similarly for the sets $A^{2,k}$ and $B^{2,k}$ in the second row. 89

2-8 The 6-stencil triangle selection independent branching formulation. The sets $A^{DL,\alpha}$ (resp. $B^{DL,\alpha}$) are the squares (resp. diamonds) in the first row; similarly for the sets $A^{ADL,\alpha}$ and $B^{ADL,\alpha}$ in the second row. The diagonal/antidiagonal lines considered in each level are circled. 89

2-9 Visualizations of the biclique cover from the proof of Theorem 7 for SOS3(26). Each row corresponds to some level j, and the sets A^j and B^j are the blue squares and green diamonds, respectively. The first three rows correspond the the "first stage" of the biclique cover $\{(A^{1,j}, B^{1,j})\}_{j=1}^3$, and the second nine correspond to the "second stage" $\{(A^{2,j}, B^{2,j})\}_{j=1}^9$. 92

3-1 Two embeddings with the encoding $H = (1, 2, 3)$ and different orders of the sets as **(Left)** $\mathcal{P} = ([0, 1], [2, 3], [4, 5])$ and **(Right)** $\tilde{\mathcal{P}} = ([2, 3], [0, 1], [4, 5])$. The ordering $\mathcal{P}$ satisfies (3.1); the ordering $\tilde{\mathcal{P}}$ does not, as can be seen from the slice at $h = 2$. 101

3-2 A family $\mathcal{P} = (P^i \subset \mathbb{R})_{i=1}^{8}$ for which the one-dimensional encoding $H = (k)_{k=1}^{8} \subset \mathbb{Z}$ is such that $Q(\mathcal{P}, H)$ is a valid MIP formulation for $\bigcup_{i=1}^{8} P^i$. **(Top)** A depiction of the embedding $\mathrm{Em}(\mathcal{P}, H)$ and its convex hull $Q(\mathcal{P}, H)$. **(Bottom)** The disjunctive set $\bigcup_{i=1}^{8} P^i$ on the real line. 102

3-3 A family $\mathcal{P} = (P^i \subset \mathbb{R})_{i=1}^{16}$, along with an encoding $H = (h^i)_{i=1}^{16} \subseteq \mathbb{Z}^2$ of dimension 2. **(Top)** A depiction of H, the associated set P^i with each code h^i, and the convex hull $\mathrm{Conv}(H)$ of the codes. **(Bottom)** The disjunctive set $\bigcup_{i=1}^{16} P^i$ on the real line. 103

3-4 **(Left)** As $\dim(L) = 1$, we cannot tilt the inequality (given by coefficients (a, b)) to make one of the inequalities defining K binding, while maintaining feasibility with respect to L and R. **(Right)** This tilting is possible when $\dim(L) > 1$. 110

3-5 Depiction of **(Left)** H_8^{Log}, **(Center)** H_8^{ZZI}, and **(Right)** H_8^{ZZB}. The first row of each is marked with a dot, and the subsequent rows follow along the arrows. The axis orientation is different for H_8^{ZZB} for visual clarity. 115

3-6 Annulus with $d = 8$ quadrilateral pieces (crosshatched), along with LP relaxation of ideal formulation (solid light gray). 119

3-7 LP relaxation of formulation (3.8) (shaded) after **(Left)** down-branching $z_1 \leqslant 0$, and **(Right)** up-branching $z_1 \geqslant 1$. The quadrilaterals that are feasible for each subproblem are crosshatched. 120

3-8 LP relaxation of formulation (3.9) (shaded) after **(First row)** down-branching $z_1 \leqslant 0$ or up-branching $z_1 \geqslant 1$; **(Second row)** down-branching $z_1 \leqslant 1$ or up-branching $z_1 \geqslant 2$; **(Third row)** down-branching $z_1 \leqslant 2$ or up-branching $z_1 \geqslant 3$; or **(Last row)** down-branching $z_1 \leqslant 3$ or up-branching $z_1 \geqslant 4$. The quadrilaterals that are feasible for each subproblem are crosshatched. 121

3-9 LP relaxation of embedding formulation for the annulus using the binary zig-zag encoding H_8^{ZZB} (shaded) after **(Left)** down-branching $z_1 \leqslant 0$, and **(Right)** up-branching $z_1 \geqslant 1$. The quadrilaterals that are feasible for each subproblem are crosshatched. 122

4-1 The LP relaxation of an ideal formulation for (1.11) (e.g. `Log`/`LogIB`, `Inc`, or `ZZI`) projected onto (x, y)-space. 130

4-2 The LP relaxation of the `Inc` formulation (4.3) projected onto (x, y)-space, after **(Top Left)** down-branching $z_1 \leqslant 0$, **(Top Right)** up-branching $z_1 \geqslant 1$; **(Center Left)** down-branching $z_2 \leqslant 0$, **(Center Right)** up-branching $z_2 \geqslant 1$; **(Bottom Left)** down-branching $z_3 \leqslant 0$, and **(Bottom Right)** up-branching $z_3 \geqslant 1$. 132

4-3 The `Log` formulation after **(Left)** down-branching $z_1 \leqslant 0$ and **(Right)** up-branching $z_1 \geqslant 1$. 133

4-4 The LP relaxation of the `ZZI` formulation (4.2) projected onto (x, y)-space, after **(Top Center)** down-branching $z_1 \leqslant 0$, **(Bottom Center)** up-branching $z_1 \geqslant 1$, **(Top Right)** down-branching $z_1 \leqslant 1$, and **(Bottom Right)** up-branching $z_1 \geqslant 2$. 134

4-5 `PiecewiseLinearOpt` code showing how to add univariate and bivariate piecewise linear functions to a JuMP model. 143

4-6 JuMP code for the cutting circle problem using `PiecewiseLinearOpt`. 145

5-1 Illustration of the branching scheme for the moment curve encoding H_7^{mc}. The original code relaxation in the z-space is shown in the dashed region, and those for the two subproblems are shown in the darker shaded regions. The optimal solution for the original LP relaxation is depicted with a solid dot. We show the branching with a solution $\hat{z}$ that is fractional **(Left)**, and one where $\hat{z} \in \mathrm{Conv}(H_7^{\mathrm{mc}}) \backslash H_7^{\mathrm{mc}}$ and yet there is no valid variable branching disjunction to separate the point **(Right)**. 165

5-2 The exotic two-dimensional encoding H_{16}^{ex}. 166

5-3 Branching scheme for the exotic encoding H^{mc}_{16} when the LP optimal solution for the integer variables $\hat{z}$ has: **(Left)** $\hat{z}_1$ fractional, **(Center)** $\hat{z}_1 \in \mathbb{Z}$ but $\hat{z}_2 \notin Y = \{ h_2 \mid h \in H^{\text{mc}}_{16} \}$, and **(Right)** $\hat{z}_1 \in \mathbb{Z}$, $\hat{z}_2 \in Y$, and yet $\hat{z} \notin H^{\text{mc}}_{16}$. The relaxations for the two subproblems in each are the two shaded regions in each picture. 168

5-4 A grid triangulation with $d = 8$ triangles. The nodes, or vertices for the triangles, are numbered. 173

5-5 The LP relaxation of the moment curve formulation (5.8) applied to the the piecewise linear function (1.11) projected onto (x, y)-space, after **(Top Left)** branching on $\Psi_4(1, 1)$, **(Top Right)** branching on $\Psi_4(2, 4)$; **(Center Left)** branching on $\Psi_4(1, 2)$, **(Center Right)** branching on $\Psi_4(3, 4)$; **(Bottom Left)** branching on $\Psi_4(1, 3)$, and **(Bottom Right)** branching on $\Psi_4(4, 4)$. 175

5-6 The LP relaxation of the moment curve formulation (5.8) projected onto (x, y)-space, after **(Top Left)** down-branching $z_1 \leqslant 1$, **(Top Right)** up-branching $z_1 \geqslant 2$; **(Center Left)** down-branching $z_1 \leqslant 2$, **(Center Right)** up-branching $z_1 \geqslant 3$; **(Bottom Left)** down-branching $z_1 \leqslant 3$, and **(Bottom Right)** up-branching $z_1 \geqslant 4$. 176

5-7 The LP relaxation of the very small SOS2 MIP-with-holes formulation (5.9) projected onto (x, y)-space, after **(Top Left)** down-branching $z_1 \leqslant -1$, **(Top Right)** up-branching $z_1 \geqslant 0$; **(Center Left)** down-branching $z_1 \leqslant 0$, and **(Center Right)** up-branching $z_1 \geqslant 1$. 178

5-8 LP relaxation of the exotic formulation (5.11) (shaded) after **(First row)** down-branching $z_1 \leqslant -2$ or up-branching $z_1 \geqslant -1$; **(Second row)** down-branching $z_1 \leqslant -1$ or up-branching $z_1 \geqslant 0$; **(Third row)** down-branching $z_1 \leqslant 0$ or up-branching $z_1 \geqslant 1$; or **(Last row)** down-branching $z_1 \leqslant 1$ or up-branching $z_1 \geqslant 2$. The quadrilaterals that are feasible for each subproblem are crosshatched. 180

B-1 The joining or splitting which transform a polygonal partition into one adhering to the conditions in Theorem 2 (before on Top, after on Bottom): **(Left)** splitting along an internal vertex, **(Center)** filling in a minimal infeasible set T of cardinality three with $\mathrm{Conv}(T) \subset \Omega$, and **(Right)** introducing an artificial vertex to remove a minimal infeasible set T of cardinality three with $\mathrm{Conv}(T) \not\subset \Omega$. 184

C-1 Feasible region in the (x, y)-space for the `Log` formulation (C.2) after: **(Left)** down-branching $z_1 \leqslant 0$, and **(Right)** up-branching $z_1 \geqslant 1$. . . . 189

C-2 Feasible region in the (x, y)-space for the `ZZI` formulation (C.3) after: **(Top first column)** down-branching on $z_1 \leqslant 0$, **(Bottom first column)** up-branching on $z_1 \geqslant 1$; **(Top second column)** down-branching on $z_1 \leqslant 1$, **(Bottom second column)** up-branching on $z_1 \geqslant 2$; **(Top third column)** down-branching on $z_1 \leqslant 2$, **(Bottom third column)** up-branching on $z_1 \geqslant 3$; **(Top fourth column)** down-branching on $z_1 \leqslant 3$, and **(Bottom fourth column)** up-branching on $z_1 \geqslant 4$. 189

List of Tables

1.1 Notation used throughout the paper. 49

4.1 Computational results for univariate transportation problems on small networks with non powers-of-two segments. 126

4.2 Computational results with Gurobi for univariate transportation problems on small networks with non powers-of-two segments. 127

4.3 Computational results for univariate transportation problems on large networks with non powers-of-two segments. 127

4.4 Difficult univariate transportation problems on large networks with non powers-of-two segments. 127

4.5 Metrics for each possible branching decision on z_1 for `Inc`, `Log`, and `ZZI` applied to (1.11). Notationally, $1 \downarrow$ means down-branching on the value 1 (i.e. $z_1 \leqslant 1$, and similarly $2 \uparrow$ implies up-branching on the value 2 (i.e. $z_1 \geqslant 2$. 131

4.6 Computational results for bivariate transportation problems on grids of size $= d_1 = d_2$. 139

4.7 Computational results for transportation problems whose objective function is the sum of bivariate piecewise linear objective functions on grids of size $\kappa = d_1 = d_2$, when κ is not a power-of-two. 140

4.8 Comparison of bivariate piecewise-linear grid triangulation formulations using the 6-stencil approach (6S) against an optimal triangle selection formulation (Opt) on grids of size $\kappa = d_1 = d_2$. 141

C.1 Metrics for each possible branching decision on z_1 for `Log` and `ZZI` applied to (C.1). 189

D.1 Computational results with Gurobi for univariate transportation problems on large networks with non powers-of-two segments. 191

D.2 Computational results with Gurobi for bivariate transportation problems on grids of size $\kappa = d_1 = d_2$. 192

Chapter 1

Preliminaries.

The aim of this thesis is to develop new methods to solve difficult optimization problems, efficiently. The class of "difficult" optimization problems we focus on are nonconvex, and possibly discrete. Nonconvex optimization is difficult from a complexity perspective, even for restrictive subclasses such as nonconvex quadratic optimization [112]. Therefore, we mean "efficiently" in a practical sense, and we will endeavor to produce provably optimal solutions (or rigorous bounds on suboptimality) in a reasonable amount of time. At the end of this chapter, we will present a taxonomy of nonconvex structures that arise in a number of compelling applications. After this, we spend the rest of this thesis developing a number of advanced techniques to build good representations, or *formulations*, for them, and apply these techniques to produce substantial speed-ups over existing approaches.

1.1 Mixed-integer programming and formulations

Suppose that we have an optimization problem of the form

$$\min_{x,y} \quad b \cdot x + c \cdot y \tag{1.1a}$$

$$\text{s.t.} \quad x \in D \tag{1.1b}$$

$$(x, y) \in X, \tag{1.1c}$$

where X is a convex set, and D is a nonconvex set that describes some portion of the feasible region.

In this context, it is well known that merely convexifying D by replacing it with its convex hull (i.e. changing (1.1b) to $x \in \text{Conv}(D)$) is not sufficient for solving (1.1)[1]. Instead, we will add auxiliary variables w and z and craft a *linear programming (LP) relaxation* given by an inequality (outer) description as

$$R = \left\{ (x, w, z) \in \mathbb{R}^{n+p+r} \;\middle|\; \begin{array}{c} l^x \leqslant x \leqslant u^x \\ l^w \leqslant w \leqslant u^w \\ l^z \leqslant z \leqslant u^z \\ Ax + Bw + Cz \leqslant d \end{array} \right\}. \tag{1.2}$$

We will say say that

$$F = \{ (x, y, z) \in R \mid z \in \mathbb{Z}^r \} \tag{1.3}$$

is a *mixed-integer programming (MIP) formulation* for D if

$$D = \text{Proj}_x(F) \overset{\text{def}}{=} \{ x \mid \exists w, z \text{ s.t. } (x, w, z) \in F \}.$$

We call x the *original variables*, w the (optional) *continuous auxiliary variables*, and z the *integer variables* of our formulation. An important subclass of formulations will be *binary* MIP formulations, where $l^z = \mathbf{0}^r$ and $u^z = \mathbf{1}^r$. Otherwise, we call F a *general integer* MIP formulation.

We can use our formulation to solve the optimization problem (1.1) as

$$\min_{x,y,z,w} \quad b \cdot x + c \cdot y \tag{1.4a}$$

$$\text{s.t.} \quad (x, w, z) \in R \tag{1.4b}$$

$$(x, y) \in X \tag{1.4c}$$

$$z \in \mathbb{Z}^r. \tag{1.4d}$$

[1] As a simple example, take $b = c = 1$, $D = \{-1, 1\}$, and $X = \mathbb{R}^2_{\geqslant 0}$. The optimal solution to (1.1) is $(x, y) = (1, 0)$ with cost 1, while the convexified version has optimal solution $(x, y) = (0, 0)$ with cost 0.

This is a mixed-integer programming problem, with a convex relaxation and nonconvexity arising only from the integrality imposed on the z variables. Of course, realistic optimization problems may contain a number of nonconvexities, meaning we will need to repeat this procedure.

Mixed-integer programming is surprisingly expressive class of optimization problems, capable of modeling many complex problems of interest throughout operations [38, 39, 90], analytics [18, 19], engineering [56, 60, 124], and robotics and control [15, 46, 47, 49, 85, 103, 114], to name just a few. Moreover, there exist sophisticated algorithms–and corresponding high-quality software implementations–that can solve many problems of this form efficiently in practice [24, 74]. For the remainder, we will focus on building formulations for nonconvex substructures like D individually, and then composing them afterwards as in (1.4).

1.2 Disjunctive constraints

A central modeling primitive in mathematical optimization is the disjunctive constraint: any feasible solution must satisfy at least one of some fixed, finite collection of alternatives. This type of constraint is general enough to capture structures as diverse as boolean satisfiability, complementarity constraints, special-ordered sets, and (bounded) integrality. The special case of polyhedral disjunctive constraints corresponds to the form

$$x \in D \overset{\mathsf{def}}{=} \bigcup_{i=1}^{d} P^i, \tag{1.5}$$

where we have that each *alternative* $P^i \subseteq \mathbb{R}^n$ is a polyhedron. We will make the following simplifying assumption on the structure of D.

Assumption 1. *There are a finite number of alternatives P^i, and each is a rational, bounded polyhedron.*

However, we mention below ways to extend our results to the unbounded case.

1.2.1 Combinatorial disjunctive constraints

We will spend much of this thesis focusing on a particular class of disjunctive constraints that are both incredibly expressive and readily amenable to advanced formulation construction techniques. In particular, we will focus on disjunctive constraints where each alternative is a face of the standard simplex.

Definition 1. *Take some finite set $V \subset \mathbb{R}^n$. A* combinatorial disjunctive constraint *(CDC) given by the family of sets $\mathcal{T} = (T^i \subseteq V)_{i=1}^d$ is a disjunctive constraint of the form*

$$\lambda \in \operatorname{CDC}(\mathcal{T}) \stackrel{\text{def}}{=} \bigcup_{i=1}^{d} P(T^i),$$

where

- $\operatorname{supp}(\lambda) \stackrel{\text{def}}{=} \{ v \in V \mid \lambda_v \neq 0 \}$ *is the set of nonzero components of* λ*,*
- $\Delta^V \stackrel{\text{def}}{=} \left\{ \lambda \in \mathbb{R}^V_{\geqslant 0} \,\middle|\, \sum_{v \in V} \lambda_v = 1 \right\}$ *is the standard simplex, and*
- $P(T) \stackrel{\text{def}}{=} \left\{ \lambda \in \Delta^V \,\middle|\, \operatorname{supp}(\lambda) \subseteq T \right\}$ *is the face that $T \subseteq V$ induces on the standard simplex.*

We call V the ground set *for the constraint.*

Although combinatorial disjunctive constraints can arise naturally as primitive constraints (see Chapters 1.3.3 and 1.3.4), they also offer a principled way to construct formulations for arbitrary disjunctive constraints.

Due to the celebrated Minkowski-Weyl Theorem (e.g. [34, Corollary 3.14]), we can write each of our polyhedral alternatives P^i as the convex combination of its extreme points $T^i = \operatorname{ext}(P^i)$:

$$P^i = \operatorname{Conv}(T^i) \stackrel{\text{def}}{=} \left\{ \sum_{v \in T^i} \lambda_v v \,\middle|\, \lambda \in \Delta^{T^i} \right\}. \tag{1.6}$$

Suppose we add additional components to the convex multipliers λ, so that $\lambda \in \Delta^V$

for $V = \bigcup_{i=1}^d \operatorname{ext}(P^i)$. Then (1.6) is equivalent to

$$P^i = \left\{ \sum_{v \in V} \lambda_v v \;\middle|\; \lambda \in P(T^i) \right\},$$

where the constraint $\lambda \in P(T^i)$ ensures that each new component of λ not corresponding to an extreme point of P^i must be zero. If we take the family of sets $\mathcal{T} = (T^i = \operatorname{ext}(P^i))_{i=1}^d$, then the combinatorial disjunctive constraint approach gives us a way to formulate a disjunctive set as

$$D \equiv \bigcup_{i=1}^d P^i = \left\{ \sum_{v \in V} \lambda_v v \;\middle|\; \lambda \in \operatorname{CDC}(\mathcal{T}) \right\}. \tag{1.7}$$

Note that all the nonconvexity of the set D has now been encapsulated in $\operatorname{CDC}(\mathcal{T})$. Therefore, we can focus on formulating $\operatorname{CDC}(\mathcal{T})$, which will often be much simpler than formulating D directly, and then easily construct a formulation for D by applying a linear transformation to the λ variables. The disjunctive constraints are combinatorial since they rely on the shared structure of the extreme points among the different alternatives, captured in the family of sets $\mathcal{T}$.

Moreover, it is straightforward to extend the combinatorial disjunctive constraint approach to accommodate unbounded alternatives (provided standard representability conditions are satisfied [72] [130, Proposition 11.2]) as

$$D = \left\{ \sum_{v \in V} \lambda_v v + \sum_{r \in R} \mu_r r \;\middle|\; \lambda \in \operatorname{CDC}(\mathcal{T}),\ \mu \in \mathbb{R}^R_{\geqslant 0} \right\}, \tag{1.8}$$

where R is the shared set of extreme rays for each of the P^i. We offer this as justification for Assumption 1, as formulating the unbounded case is a straightforward extension.

Much of the work of this thesis will be in deriving formulations for $\operatorname{CDC}(\mathcal{T})$, given a particular family of sets $\mathcal{T}$ over some ground set V. We make the following assumptions about $\mathcal{T}$ that are without loss of generality.

Assumption 2. *We assume the following about* $\mathcal{T} = (T^i \subseteq V)_{i=1}^d$.

- *The constraint is disjunctive:* $|\mathcal{T}| > 1$.
- *Each alternative is nonempty:* $T \neq \emptyset$ *for all* $T \in \mathcal{T}$.
- $\mathcal{T}$ *is irredundant: there do not exist distinct* $T, T' \in \mathcal{T}$ *such that* $T \subseteq T'$.
- $\mathcal{T}$ *covers the ground set:* $\bigcup_{i=1}^{d} T^i = V$.

We will say that a set $T \subseteq V$ is a *feasible set* with respect to (w.r.t.) $\mathrm{CDC}(\mathcal{T})$ if $P(T) \subseteq \mathrm{CDC}(\mathcal{T})$ (equivalently, if $T \subseteq T'$ for some $T' \in \mathcal{T}$) and that it is an *infeasible set* otherwise. Notationally, given the family of sets $\mathcal{T} = (T^i \subseteq V)_{i=1}^{d}$, we will find it useful to refer to the corresponding family of faces of the standard simplex $\mathcal{P}(\mathcal{T}) \stackrel{\text{def}}{=} (P(T^i))_{i=1}^{d}$, which are the alternatives of the disjunctive constraint which we ultimately will formulate.

1.2.2 Combinatorial disjunctive constraints and data independence

One distinct advantage of the combinatorial disjunctive constraint approach is that the formulation (1.7) allows us to divorce the problem-specific data (i.e. the values $v \in V$) from the underlying combinatorial structure in $\mathcal{T}$. As such, we can construct a single, strong formulation for a given structure $\mathrm{CDC}(\mathcal{T})$, and this formulation will remain valid for transformations of the data, so long as this transformation sufficiently preserves the combinatorial structure of $\mathrm{CDC}(\mathcal{T})$. For instance, if we formulate $\bigcup_{i=1}^{d} P^i$ via its corresponding combinatorial disjunctive constraint $\mathrm{CDC}(\mathcal{T})$, and then change the data to produce a related disjunctive constraint $\bigcup_{i=1}^{d} \hat{P}^i$, a sufficient condition for our formulation of $\mathrm{CDC}(\mathcal{T})$ to yield a valid formulation for $\bigcup_{i=1}^{d} \hat{P}^i$ is the existence of a bijection $\pi : V \to \hat{V}$ (with $V = \bigcup_{i=1}^{d} \mathrm{ext}(P^i)$ and $\hat{V} = \bigcup_{i=1}^{d} \mathrm{ext}(\hat{P}^i)$) such that

$$v \in \mathrm{ext}(P^i) \iff \pi(v) \in \mathrm{ext}(\hat{P}^i) \quad \forall i \in [\![d]\!],\ v \in V, \tag{1.9}$$

where $[\![d]\!] \stackrel{\text{def}}{=} \{1, \ldots, d\}$. In this way, we can construct a single small, strong formulation for $\mathrm{CDC}(\mathcal{T})$, and use it repeatedly for many different "combinatorially equivalent" instances of the same constraint.

We note that one subtle disadvantage of this data-agnostic approach is that, even if condition (1.9) is satisfied, the resulting formulation for $\bigcup_{i=1}^d \hat{P}^i$ may be larger than necessary. An extreme manifestation of this would be when the new polyhedra $(\hat{P}^i)_{i=1}^d$ are such that $\hat{P}^i \subseteq \hat{P}^1$ for all $i \in [\![d]\!]$. In this case, $\bigcup_{i=1}^d \hat{P}^i = \hat{P}^1$ and the constraint is no longer truly disjunctive, meaning it can be modeled directly as a LP. Less pathological cases could occur where some subset of the disjunctive sets become redundant after changing the problem data. However, we note that in many of the applications considered in this work, the combinatorial representation leads to redundancy of this form only in rare pathological cases (this is true of the constraints in Chapter 1.3.2, for example). In other cases we will take care to consider, for example, the geometric structure of the data before constructing the disjunctive constraint (e.g. our results in Chapter 2.3.3).

1.3 Motivating problems

We now present constraints, arising in a number of application areas across engineering, robotics, power systems, and machine learning, that we will return to throughout. The first seven will fall neatly into the combinatorial disjunctive constraint framework. However, the last does not, and we will explore other techniques to construct strong formulations for it in Chapter 4.

1.3.1 Modeling discrete alternatives

The special-ordered set (SOS) constraints introduced by Beale and Tomlin [13] are a classical family of constraints with numerous applications throughout operations research. The SOS constraint of type 1 (SOS1) is given by the family of sets $\mathcal{T}_N^{\text{SOS1}} \stackrel{\text{def}}{=} (\{i\})_{v=1}^N$, and can be used to model discrete alternatives: given N distinct points $\{v^i\}_{i=1}^N \subset \mathbb{R}^n$,

$$x \in \{v^i\}_{i=1}^N \iff x = \sum_{i=1}^N \lambda_i v^i \text{ for some } \lambda \in \text{CDC}(\mathcal{T}_N^{\text{SOS1}}).$$

Notationally, we will refer to the SOS1 constraint on N components (i.e. given by $\mathcal{T}_N^{\text{SOS1}}$ with ground set $V = [\![N]\!]$) as SOS1(N). In the following subsections, we will present other types of SOS constraints.

1.3.2 Piecewise linear functions

Consider an optimization problem of the form $\min_{x \in \Omega} f(x)$, where $\Omega \subseteq \mathbb{R}^n$ and $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ is a piecewise linear function. That is, f can be described by a partition of the domain Ω into a finite family $\{C^i\}_{i=1}^d$ of polyhedral pieces, where for each piece C^i there is an affine function $f^i : C^i \to \mathbb{R}$ where $f(x) = f^i(x)$ for each $x \in C^i$. In the same vein, we may consider an optimization problem where the feasible region is (partially) defined by a constraint of the form $f(x) \leqslant 0$, where f is piecewise linear.

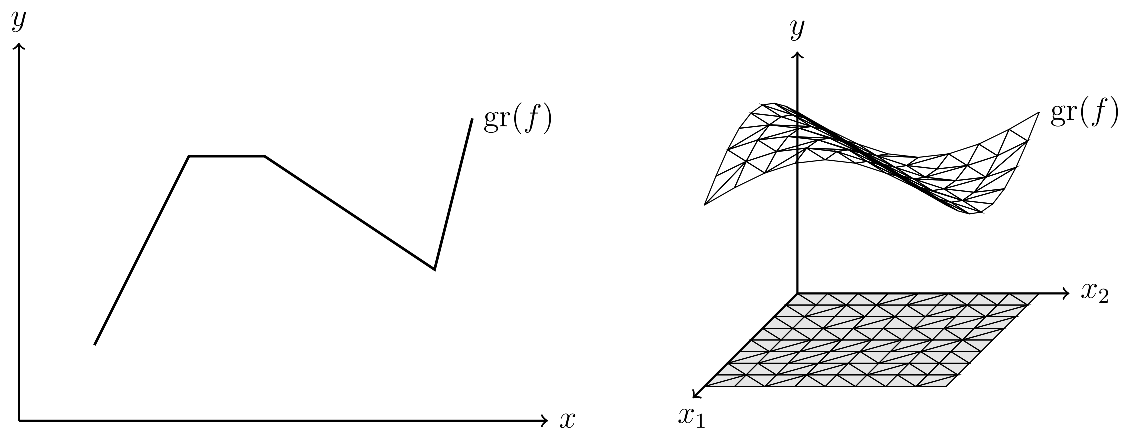


Figure 1-1: **(Left)** A univariate piecewise linear function, and **(Right)** a bivariate piecewise linear function with a grid triangulated domain.

The potential applications for this class of optimization problems are legion. Piecewise linear functions arise naturally throughout operations [38, 39, 90] and engineering [56, 60, 124]. They are a natural choice for approximating nonlinear functions, as they often lead to optimization problems that are easier to solve than the original problem [16, 17, 28, 58, 82, 106, 104]. For example, there has been recently been significant interest in using piecewise linear functions to approximate complex nonlinearities arising gas network optimization problems [32, 33, 101, 97, 105, 125]; see [80] for a recent book on the subject.

If the function f happens to be convex, it is possible to reformulate our optimization problem into an equivalent LP problem (provided that Ω is polyhedral) However, if f is nonconvex, this problem is NP-hard in general [78]. A number of specialized algorithms for solving piecewise linear optimization problems have been proposed over the years [13, 45, 44, 78, 128]. Instead, we will focus on MIP formulations for piecewise linear function, an active and fruitful area of research for decades [10, 38, 41, 42, 72, 71, 77, 89, 96, 99, 111, 123, 135, 133, 137].

We consider continuous[2] piecewise linear functions $f : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^n$ is bounded. We will describe f in terms of the domain pieces $\{C^i \subseteq \Omega\}_{i=1}^d$ and the corresponding affine functions $\{f^i\}_{i=1}^d$. We assume that the pieces cover the domain Ω, and that their relative interiors do not overlap. Furthermore, we assume that our function f is *non-separable* and cannot be decomposed as the sum of lower-dimensional piecewise linear functions. This is without loss of generality, as if such a decomposition exists, we could apply our formulation techniques to the individual pieces separately. Finally, we will spend a substantial amount time on the regime where the dimension n of the domain is relatively small: when f is either univariate ($n = 1$) or bivariate ($n = 2$) with a grid triangulated domain; see Figure 1-1 for an illustrative example of each.

In order to solve an optimization problem containing f, we will construct a formulation for its graph

$$\operatorname{gr}(f;\Omega) \stackrel{\text{def}}{=} \{\, (x, f(x)) \mid x \in \Omega \,\}.$$

We can write this graph disjunctively as $\operatorname{gr}(f) = \bigcup_{i=1}^d P^i$, where each alternative $P^i = \{\, (x, f^i(x)) \mid x \in C^i \,\}$ is a *segment* of the graph. We can then take $\mathcal{T} = (\operatorname{ext}(C^i))_{i=1}^d$, formulate $\operatorname{CDC}(\mathcal{T})$, and express the graph as

$$\operatorname{gr}(f) = \left\{ \sum_{v \in V} \lambda_v (v, f(v)) \;\middle|\; \lambda \in \operatorname{CDC}(\mathcal{T}) \right\}, \tag{1.10}$$

where we will use the notation $\operatorname{gr}(f) \equiv \operatorname{gr}(f;\Omega)$ when Ω is clear from context.

[2]The results to follow can potentially be extended to discontinuous piecewise linear functions by working instead with the epigraph of f; we point the interested reader to [133, 134].

Univariate piecewise linear functions

For univariate piecewise linear functions (i.e. $\Omega \subset \mathbb{R}$), the corresponding combinatorial disjunctive constraint has a particularly nice structure. The domain partition will consist of d adjacent intervals $C^i = [\tau^i, \tau^{i+1}]$, given by $N \equiv d+1$ breakpoints $\tau^1 < \tau^2 < \cdots < \tau^N$ that we presume are distinct. In particular, the family of sets will be equivalent to the special-ordered set of type 2 (SOS2) constraint, which is defined as $\mathcal{T}_d^{\text{SOS2}} \stackrel{\text{def}}{=} (\{i, i+1\})_{i=1}^d$. In words, the constraint $\lambda \in \text{CDC}(\mathcal{T}_d^{\text{SOS2}})$ requires that at most two components of λ may be nonzero, and that these nonzero components must be consecutive in the ordering on $V = \llbracket N \rrbracket$. We will refer to the SOS2 constraint on $N \equiv d+1$ components as SOS2(N).

As long as the ordering of the breakpoints is preserved, it is easy to see that condition (1.9) will be satisfied for any transformations of the problem data. Furthermore, the only case in which knowledge of the specific data $\{(\tau^i, f(\tau^i))\}_{i=1}^d$ allows the simplification of the original disjunctive representation of $\text{gr}(f)$ is when f is affine in on two adjacent intervals, e.g. affine over $[\tau^i, \tau^{i+2}]$ for some $i \in \llbracket d-1 \rrbracket$. Therefore, the potential disadvantage of disregarding the specific data when formulating the constraint occurs only in rare pathological cases which are easy to detect.

Example 1 (A univariate piecewise linear function)**.** *Consider the univariate piecewise linear function $f : [0,4] \to \mathbb{R}$. We decompose the domain into the four pieces $C^1 = [0,1]$, $C^2 = [1,2]$, $C^3 = [2,3]$, and $C^4 = [3,4]$, and describe the function as*

$$x \in C^1 \implies f(x) = 4x, \qquad x \in C^2 \implies f(x) = 3x+1, \tag{1.11a}$$

$$x \in C^3 \implies f(x) = 2x+3, \qquad x \in C^4 \implies f(x) = x+6. \tag{1.11b}$$

The graph of the piecewise linear function is then

$$\begin{aligned} \text{gr}(f) = \left\{ (x, 4x) \;\middle|\; x \in C^1 \right\} \cup \left\{ (x, 3x+1) \;\middle|\; x \in C^2 \right\} \cup \\ \left\{ (x, 2x+3) \;\middle|\; x \in C^3 \right\} \cup \left\{ (x, x+6) \;\middle|\; x \in C^4 \right\}. \end{aligned}$$

Moreover, the function has $d = 4$ segments, and is given by the breakpoints $\{\tau^i =$

$(i-1)\}_{i=1}^{4}$. *Taking, without loss of generality (w.l.o.g.), the ground set as* $V = [\![5]\!]$, *we can observe that*

$$(x,y) \in \mathrm{gr}(f) \iff (x,y) = \sum_{v \in V} (\tau^v, f(\tau^v))\lambda_v \textit{ for some } \lambda \in \mathrm{CDC}(\mathcal{T}_4^{\mathrm{SOS2}}).$$

Bivariate piecewise linear functions and grid triangulations

Consider a (potentially nonconvex) region $\Omega \subset \mathbb{R}^2$ in the plane. We would like to model a (also potentially nonconvex) piecewise linear function f with domain over Ω; see Figure 1-1 for an illustration. If we take $\{C^i\}_{i=1}^d$ as a partition of the domain Ω, along with the family of sets $\mathcal{T} = (\mathrm{ext}(C^i))_{i=1}^d$ with ground set $V = \bigcup_{i=1}^d \mathrm{ext}(C^i)$, we can model the piecewise linear function via the graph representation (1.10).

Example 2 (A bivariate piecewise linear function)**.** *Take the bivariate piecewise linear function* $f : [0,1]^2 \to \mathbb{R}$ *given by the domain partition* $C^1 = \{ x \in [0,1]^2 \mid x_1 \geqslant x_2 \}$ *and* $C^2 = \{ x \in [0,1]^2 \mid x_1 \leqslant x_2 \}$, *and*

$$x \in C^1 \implies f(x) = -x_1 + 3x_2 + 1, \qquad x \in C^2 \implies f(x) = x_1 + x_2 + 1. \quad (1.12)$$

See the left side of Figure 1-3 for an illustration. The corresponding graph is

$$\mathrm{gr}(f) = \left\{ (x, -x_1 + 3x_2 + 1) \;\middle|\; x \in C^1 \right\} \cup \left\{ (x, x_1 + x_2 + 1) \;\middle|\; x \in C^2 \right\}.$$

Furthermore, $V = \{0,1\}^2$, *and with* $\mathcal{T} = (\{(0,0),(1,0),(1,1)\}, \{(0,0),(0,1),(1,1)\})$,

$$(x,y) \in \mathrm{gr}(f) \iff (x,y) = \sum_{v \in V} (v, f(v))\lambda_v \textit{ for some } \lambda \in \mathrm{CDC}(\mathcal{T}).$$

An important special case we will focus on occurs when the function f is affine over a *grid triangulation*, as in Figure 1-1 and Example 2. Consider a rectangular region in the plane $\Omega = [1, N_1] \times [1, N_2]$ where along each axis we apply a discretization with $d_1 = N_1 - 1$ and $d_2 = N_2 - 1$ breakpoints, respectively. This leads to a set of regular grid points $V = \{1, \ldots, N_1\} \times \{1, \ldots, N_2\}$. A *grid triangulation* $\mathcal{T}$ of Ω is then a family of sets $\mathcal{T}$ where:

- Each $T \in \mathcal{T}$ is a triangle: $|T| = 3$.
- $\mathcal{T}$ partitions Ω: $\bigcup_{T \in \mathcal{T}} \text{Conv}(T) = \Omega$ and $\text{int}(\text{Conv}(T)) \cap \text{int}(\text{Conv}(T')) = \emptyset$ for each distinct $T, T' \in \mathcal{T}$ (where $\text{int}(S)$ is the interior of set S).
- $\mathcal{T}$ is on a regular grid: $||v - w||_\infty \leqslant 1$ for each $T \in \mathcal{T}$ and $v, w \in T$.

Grid triangulations can possess a very rich and irregular combinatorial structure; see Figure 1-2, for three different triangulations with $d_1 = d_2 = 2$. Note that any formulation constructed for a given grid triangulation can be readily applied to any other grid triangulation obtained by shifting the grid points in the plane, so long as the resulting triangulation is strongly isomorphic to, or compatible with, the original triangulation [3]. In particular, our choice of ground set $V = \{1, \ldots, N_1\} \times \{1, \ldots, N_2\}$ is without loss of generality, and we can readily adapt our formulations for grids with shifts or unequal interval lengths.

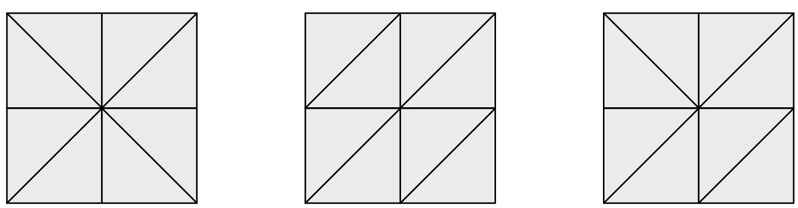


Figure 1-2: Three grid triangulations of $\Omega = [1,3] \times [1,3]$: **(Left)** the Union Jack (J1) [127], **(Center)** the K1 [84], and **(Right)** a more idiosyncratic triangulation.

Finally, we note that the choice of triangulation is important, as it materially affects the values that the corresponding piecewise linear function takes. This is in sharp contrast to the univariate case, where the function is completely determined by the breakpoints and the value the function takes at those points. In Figure 1-3 we see a simple example of this, where two different triangulations lead to two bivariate functions which coincide at the corner points V, but differ substantially in the interior of the domain.

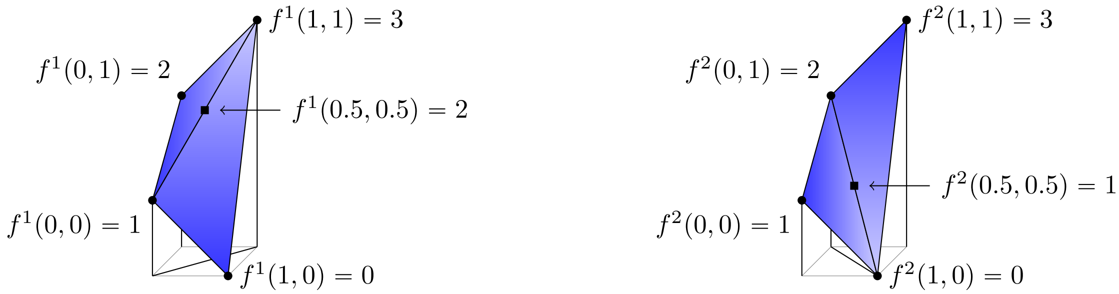


Figure 1-3: Two bivariate functions over $D = [0,1]^2$ that match on the gridpoints, but differ on the interior of D.

Higher-dimensional piecewise linear functions Beyond univariate and bivariate piecewise linear functions, it is also possible to define piecewise linear functions over higher-dimensional grid triangulations. Given a η-dimensional piecewise linear function $f : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^\eta$ is a hyperrectangular domain. As was the case with univariate and bivariate functions, assume that $\Omega = \prod_{i=1}^{\eta}[1, N_i]$ and take the regularly gridded ground set $V = \prod_{i=1}^{\eta}[\![N_i]\!]$, along with a η-dimensional grid triangulation given by the family of sets $\mathcal{T}$, where

- Each $T \in \mathcal{T}$ is a simplex: $|T| = \eta + 1$.
- $\mathcal{T}$ partitions Ω: $\bigcup_{T \in \mathcal{T}} \mathrm{Conv}(T) = \Omega$ and $\mathrm{int}(\mathrm{Conv}(T)) \cap \mathrm{int}(\mathrm{Conv}(T')) = \varnothing$ for each distinct $T, T' \in \mathcal{T}$.
- $\mathcal{T}$ is on a regular grid: $||v - w||_\infty \leqslant 1$ for each $T \in \mathcal{T}$ and $v, w \in T$.

Note that this definition is a generalization of that given in the previous subsection for grid triangulations in the plane ($\eta = 2$). The combinatorial structure for higher-dimensional grid triangulations is even more complex than in the two-dimensional case: for example, different choices of triangulation for a single hypercube (e.g. $\{0,1\}^\eta$) can have different numbers of triangles [98, 127]. One possible choice is the *standard triangulation* for $\{0,1\}^\eta$, given by the family

$$T^\pi = \left\{ x \in \{0,1\}^\eta \;\middle|\; x_{\pi(1)} \leqslant x_{\pi(2)} \leqslant \cdots \leqslant x_{\pi(\eta)} \right\} \tag{1.13}$$

for each permutation π of $[\![\eta]\!]$.

High-dimensional piecewise linear functions arise in a number of important optimization contexts (see, for example, Chapter 1.3.8), but unless η is very small they quickly strain the practicality of the combinatorial disjunctive constraint approach. This is because the cardinality of the ground set V will grow exponentially in η, and as the formulation (1.7) requires an auxiliary variable λ_v for each element $v \in V$, this overhead can quickly become overwhelming.

1.3.3 The SOSk constraint

In addition to the SOS1 and SOS2 constraints we have described above, we can generalize the class of constraints to special-ordered sets of type k (SOSk), where at most k consecutive components of λ may be nonzero at once. In particular, if $V = [\![N]\!]$, we have $\mathcal{T}^{\text{SOS}}_{N,k} \stackrel{\text{def}}{=} (\{\tau, \tau+1, \ldots, \tau+k-1\})_{i=1}^{N-k+1}$. This constraint may arise, for example, in chemical process scheduling problems, where an activated machine may only be on for k consecutive time units and must produce a fixed quantity during that period [52, 83]. We will refer to the SOSk constraint on N components as SOS$k(N)$.

1.3.4 Cardinality constraints

An extremely common constraint in optimization is the cardinality constraint of degree ℓ, where at most ℓ components of λ may be nonzero. This corresponds to the family of sets $\mathcal{T}^{\text{card}}_{N,\ell} \stackrel{\text{def}}{=} (T \subseteq V \mid |T| = \ell)$, where we presume that $V = [\![N]\!]$ for simplicity. A particularly compelling application of the cardinality constraint is in portfolio optimization [20, 22, 30, 132], where it is often advantageous to limit the number of investments to some fixed number ℓ to minimize transaction costs, or to allow differentiation from the performance of the market as a whole.

1.3.5 Discretizations of multilinear functions

Consider a multilinear function $f(x_1, \ldots, x_\eta) = \prod_{i=1}^{\eta} x_i$ defined over some hyperrectangular domain $\Omega = [l, u] \subset \mathbb{R}^\eta$. This function appears often in optimization models [54], but is nonconvex, and often leads to problems which are difficult to solve to global optimality in practice [6, 115, 136]. As a result, computational techniques will often "relax" the graph of the function $\mathrm{gr}(f)$ with a convex outer approximation, which is easier to optimize over [119].

For the bilinear case ($\eta = 2$), the well-known McCormick envelope [102] describes the convex hull of $\mathrm{gr}(f)$. Although traditionally stated in an inequality description, we may equivalently describe the convex hull via its four extreme points, which are readily available in closed form. For higher-dimensional multilinear functions, the convex hull has 2^η extreme points, and can be constructed in a similar manner (e.g. see equation (3) in [95] and the associated references).

Misener et al. [104, 106] propose a computational technique for optimizing problems with bilinear terms where, instead of modeling the graph over a single region $\Omega = [l, u] \subset \mathbb{R}^2$, they discretize the region in a regular fashion and apply the McCormick envelope to each subregion. They model this constraint as a union of polyhedra, where each subregion enjoys a tighter relaxation of the bilinear term. Additionally, they propose a logarithmically-sized formulation for the union. However, it is not ideal (see Appendix A), it only applies for bilinear terms ($\eta = 2$), and it is specialized for a particular type of discretization (namely, only discretizing along one component x_1, and with constant discretization widths).

For a more general setting, we have that the extreme points of the convex hull of the graph $\mathrm{Conv}(\mathrm{gr}(f))$ are given by $\{\, (x, f(x)) \mid x \in \mathrm{ext}(\Omega) \,\}$ [95, equation (3)], where it is easy to see that $\mathrm{ext}(\Omega) = \prod_{i=1}^{\eta}\{l_i, u_i\}$. Consider a grid imposed on $[l, u] \subset \mathbb{R}^\eta$; that is, along each component $i \in [\![\eta]\!]$, we partition $[l_i, u_i]$ along the points $l_i \equiv \tau_1^i < \tau_2^i < \cdots < \tau_{d_i}^i < \tau_{d_i+1}^i \equiv u_i$. This yields $\prod_{i=1}^{\eta} d_i$ subregions; denote them by $\mathcal{R} \stackrel{\text{def}}{=} \left\{\, R^k = \prod_{i=1}^{\eta}[\tau_{k_i}^i, \tau_{k_i+1}^i] \,\middle|\, k \in \prod_{i=1}^{\eta}[\![d_i]\!] \,\right\}$.

We can then take the polyhedral partition of Ω given by $\Phi^R \stackrel{\text{def}}{=} \mathrm{Conv}(\mathrm{gr}(f; R))$ for each subregion R, the sets as $\mathcal{T} = \left(\mathrm{ext}(\Phi^R)\right)_{R \in \mathcal{R}}$, and the ground set as $V = \bigcup\{T \in \mathcal{T}\}$. In particular, we have that $V = \prod_{i=1}^{\eta}\{\tau_1^i, \ldots, \tau_{N_i}^i\}$, where $N_i = d_i + 1$ for each i. Analogously to the notational simplification we took with grid triangulations, for the remainder we will take $V = \prod_{i=1}^{\eta}[\![N_i]\!]$ and $\mathcal{T} = (\prod_{i=1}^{\eta}\{k_i, k_i + 1\} \mid k \in \prod_{i=1}^{\eta}[\![d_i]\!])$. We also note that condition (1.9) is satisfied as long as the ordering of the discretization

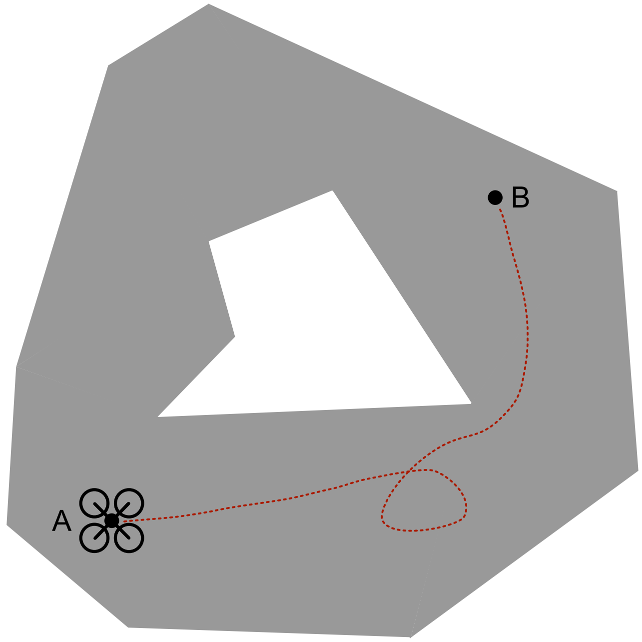


Figure 1-4: A trajectory for a UAV which avoids an obstacle in the center of the domain.

is respected along each dimension.

1.3.6 Obstacle avoidance constraints

Consider an unmanned aerial vehicle (UAV) which you would like to navigate through an area with fixed obstacles. At any given time, you wish to impose the constraint that the location of the vehicle $x \in \mathbb{R}^2$ must lie in some (nonconvex) region $\Omega \subset \mathbb{R}^2$, which is some (bounded) subset the plane, with any obstacles in the area removed. MIP formulations of this constraint has received interest as a useful primitive for path planning problems [15, 47, 103, 114].

We may model $x \in \Omega$ by partitioning the region Ω with polyhedra such that $\Omega = \bigcup_{i=1}^d P^i$. Traditional approaches to modeling constraints (1.5) of this form use a linear inequality description for each of the polyhedra P^i and construct a corresponding big-M formulation [113, 114], which will not be strong, in general. However, it is clear that we may also apply our combinatorial disjunctive constraint approach to construct formulations for obstacle avoidance constraints.

1.3.7 Annulus constraints

An annulus is a set in the plane $\mathcal{A} = \left\{ x \in \mathbb{R}^2 \;\middle|\; \underline{S} \leqslant ||x||_2 \leqslant \overline{S} \right\}$ for constants $\underline{S}, \overline{S} \in \mathbb{R}_{\geqslant 0}$; see the left side of Figure 1-5 for an illustration. A constraint of the form $x \in \mathcal{A}$ might arise when modeling a complex number $z = x_1 + x_2\mathbf{i}$, as $x \in \mathcal{A}$ bounds the magnitude of z as $\underline{S} \leqslant |z| \leqslant \overline{S}$. Such constraints arise in power systems optimization: for example, in the "rectangular formulation" [81] and the second-order cone reformulation [69, 91] of the optimal power flow problem and its voltage stability-constrained variant [40], and the reactive power dispatch problem [53]. Another application is footstep planning in robotics [46, 85], where we wish to model an angle θ trigonometrically via $x = (\cos(\theta), \sin(\theta))$. This can be accomplished by imposing the identity $x_1^2 + x_2^2 = 1$, which corresponds to taking $\underline{S} = \overline{S} = 1$.

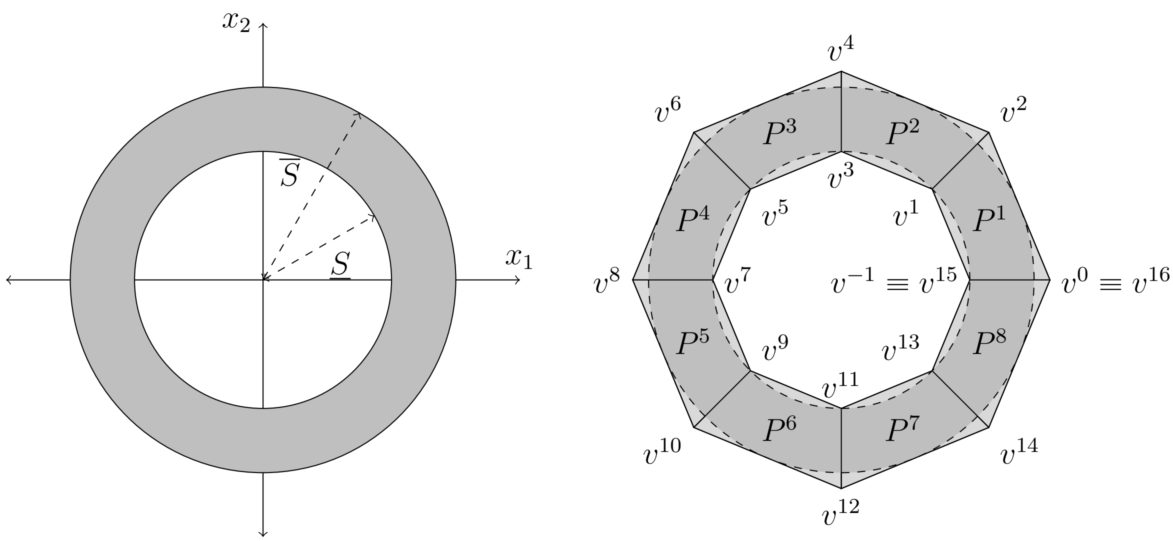


Figure 1-5: **(Left)** The annulus $\mathcal{A}$ and **(Right)** its corresponding quadrilateral relaxation $\hat{\mathcal{A}}$ given by (1.14) with $d = 8$.

When $0 < \underline{S} \leqslant \overline{S}$, $\mathcal{A}$ is a nonconvex set. Moreover, the annulus is not *mixed-integer convex representable* [93, 94]: that is, there do not exist mixed-integer formulations for the annulus even if we allow the relaxation R in formulation (1.3) to be an arbitrary convex set.

Foster [53] proposes a disjunctive relaxation for the annulus given as $\hat{\mathcal{A}} \overset{\text{def}}{=} \bigcup_{i=1}^{d} P^i$, where each

$$P^i = \text{Conv}\left(\left\{v^{2i+s-4}\right\}_{s=1}^{4}\right) \quad \forall i \in [\![d]\!] \tag{1.14}$$

is a quadrilateral based on the breakpoints

$$\begin{aligned}
v^{2i-1} &= \left(\underline{S}\cos\left(\frac{2\pi i}{d}\right), \underline{S}\sin\left(\frac{2\pi i}{d}\right)\right) && \forall i \in [\![d]\!] \\
v^{2i} &= \left(\overline{S}\sec\left(\frac{\pi}{d}\right)\cos\left(\frac{2\pi i}{d}\right), \overline{S}\sec\left(\frac{\pi}{d}\right)\sin\left(\frac{2\pi i}{d}\right)\right) && \forall i \in [\![d]\!],
\end{aligned}$$

where, for notational simplicity, we take $v^0 \equiv v^{2d}$ and $v^{-1} \equiv v^{2d-1}$. We can in turn represent this disjunctive relaxation through the combinatorial disjunctive constraint given by the family $\mathcal{T}_d^{\text{ann}} \overset{\text{def}}{=} (\{2i+s-4\}_{s=1}^4)_{i=1}^d$. See the right side of Figure 1-5 for an illustration.

1.3.8 Optimizing over trained neural networks

Since the turn of the millenia, deep learning has received an explosion of interest due to its successful application to a number of difficult problems in areas such as speech recognition and image classification [59, 87]. More recently, it has been recognized that trained feedforward neural networks with standard activation units are nothing more than high-dimensional (nonconvex) piecewise linear functions, and can be modeled using MIP. Indeed, a string of recent research has applied this observation to tasks such as verification, planning, and control [5, 31, 50, 121, 126, 138]. In this thesis, we will focus on neural networks that are built by composing a number of "rectified linear" (ReLu) activation units of the form

$$f^{\mathsf{ReLu}}(x) \overset{\text{def}}{=} \max\{0, x\}$$

with affine mappings $w \cdot x + b$ which are learned during the training procedure. In particular, we will formulate the set resulting from composing an affine mapping with a single ReLu unit:

$$\mathsf{ReLu} \stackrel{\mathsf{def}}{=} \left\{ (x, f^{\mathsf{ReLu}}(w \cdot x + b)) \;\middle|\; L \leqslant x \leqslant U \right\}.$$

As we observed in Chapter 1.3.2, the combinatorial disjunctive constraint approach does not lend itself well to high dimensional piecewise linear functions of this form, as the ground set V can grow exponentially in η. Therefore, in Chapter 4.4 we will strive to produce strong formulations through a different approach, though we will have to sacrifice formulation size as a result.

As an extension, we will also consider sets representing the composition of two layers with multiple nonlinear activation units applied to the same inputs, but with different affine mappings:

$$\left\{ (x, y^1, \ldots, y^d, \tilde{y}) \;\middle|\; \begin{array}{ll} L \leqslant x \leqslant U & \\ y^i = f^{\mathsf{ReLu}}(w^i \cdot x + b^i) & \forall i \in [\![d]\!] \\ \tilde{y} = f^{\mathsf{ReLu}}(\tilde{w} \cdot (y^1, \ldots, y^d) + \tilde{b}) & \end{array} \right\}.$$

1.4 Assessing the quality of MIP formulations

Throughout this thesis, we will be interested in ways of understanding, both quantitatively and qualitatively, when we can expect a given MIP formulation to perform well in practice. We will focus on three measures of formulation quality, which empirically tend to correlate very strongly with computational performance [34, 130, 139].

1.4.1 Strength

First, we desire formulations whose LP relaxations are as tight as possible. The reason for this is simple: using a branch-and-bound-based approach, we rely on this LP relaxation to produce good dual bounds on the optimal objective value, which we in turn use to prune as much of the search tree as possible. Therefore, we would like to produce a relaxation that is as tight as possible (without losing validity), since this will produce dual bounds that are as strong as possible.

There are two standard notions of formulation strength, that consider either the original variables x, or the integer variables z.

Definition 2. *Take a formulation* $F = \{ (x, w, z) \in R \mid z \in \mathbb{Z}^r \}$ *for* $D \subset \mathbb{R}^n$*, given by the LP relaxation* R*. We say the formulation is:*

- sharp *if* $\text{Proj}_x(R) = \text{Conv}(D)$.
- ideal *if* $\text{Proj}_z(\text{ext}(R)) \subseteq \mathbb{Z}^r$.

Sharp formulations are desirable as they offer the tightest possible convex relaxation in the x-space, and so in turn give the strongest possible dual bounds; see Figure 1-6 for an illustration. Ideal formulations are desirable since, if you solve the optimization problem over the corresponding relaxation, you are guaranteed the existence of an optimal solution that is integer feasible w.r.t. (1.4d), and optimal for the original MIP problem.

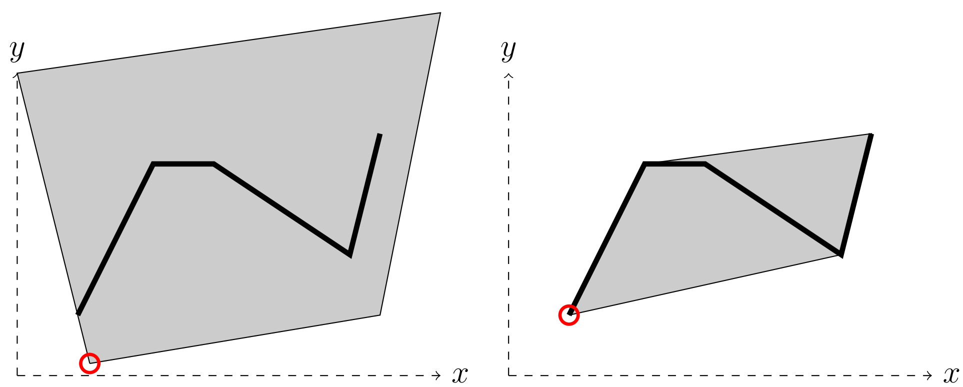


Figure 1-6: The relaxations (gray region) for two different formulations of a nonconvex set (solid lines), corresponding a univariate piecewise linear function. **(Left)** One is not sharp, while **(Right)** the second is sharp. If we solve the relaxted optimization problem $\min_{(x,y)\in R} y$ for each relaxation R, we get different optimal solution values (optimal solutions circled), and therefore different dual bounds for the MIP problem.

It is the case that an ideal formulation is also sharp, while the converse is not true [130]. Indeed, ideal formulations are the strongest possible from the perspective of their LP relaxation, hence the name. As a result, in this thesis we will focus (nearly) exclusively on ways to build ideal formulations; when we refer to "strong formulations" for the remainder, take this to mean "ideal formulations."

1.4.2 Size

Our second metric is formulation size: that is, how many variables and constraints are needed to describe the problem. We will strive to produce formulations that are as small as possible, as size (as defined below) tends to correlate very strongly with the difficulty of a MIP instance.

A first measure of formulation size is r, the number of integer variables used. This factor is of particular importance, since in the worst case the complexity of solving a MIP instance will scale exponentially in r.

Beyond this, we will count p, the number of auxiliary continuous variables, as well as the number of inequalities needed to describe the relaxation R. For our purposes we will ignore the number of original variables, as we consider them to be intrinsic to the problem. When using a branch-and-bound-based method, you will typically need to solve many optimization problems over the LP relaxation R (possibly slightly altered with new constraints or different variable bounds). Therefore, the speed at which you can solve these LPs is of tantamount importance, and so we will endeavor to produce LP relaxations that are as small as possible.

We will say that a formulation is *extended* if there are auxiliary continuous variables w in the formulation (that is, $p > 0$) and *non-extended* otherwise ($p = 0$). Furthermore, as suggested by the definition of R in (1.2), we distinguish between variable bounds (e.g. $l^x \leqslant x \leqslant u^x$) and general inequality constraints ($Ax + Bw + Cz \leqslant d$), as modern MIP solvers are able to incorporate variable bounds with minimal extra computational cost.

1.4.3 Branching behavior

Our third metric is the branching behavior of a formulation: namely, how does the LP relaxation change in a branch-and-bound algorithm? In this setting, the algorithm solved the relaxed optimization problem, producing a solution $(x^*, w^*, z^*) \in R$. It then selects a fractional integer variable z_i^* (assuming one exists) and branches on it, creating two subproblems: one with the altered variable bound $z_i \leqslant \lfloor z_i^* \rfloor$, the other with $z_i \geqslant \lceil z_i^* \rceil$. In other words, the LP relaxations for each subproblem are now $\{ (x, w, z) \in R : z_i \leqslant \lfloor z_i^* \rfloor \}$ and $\{ (x, w, z) \in R : z_i \geqslant \lceil z_i^* \rceil \}$, respectively. Ideally, both subproblem LP relaxations are substantially smaller (in a geometric sense) than R, as this will likely improve the dual bounds, and hopefully lead to substantial pruning of the search tree. However, we will see in Chapters 3 and 4 that this is often not the case: formulations can easily induce poor branching where either one or both of the subproblems do not substantially alter the LP relaxation, which can in turn lead to undesirable levels of enumeration in the search tree. See Figure 1-7 for an illustration of good and bad branching behavior: one formulation induces branching where both subproblems contract the LP relaxation substantially, while the other has very unbalanced branching, with one subproblem LP relaxation remaining completely unchanged.

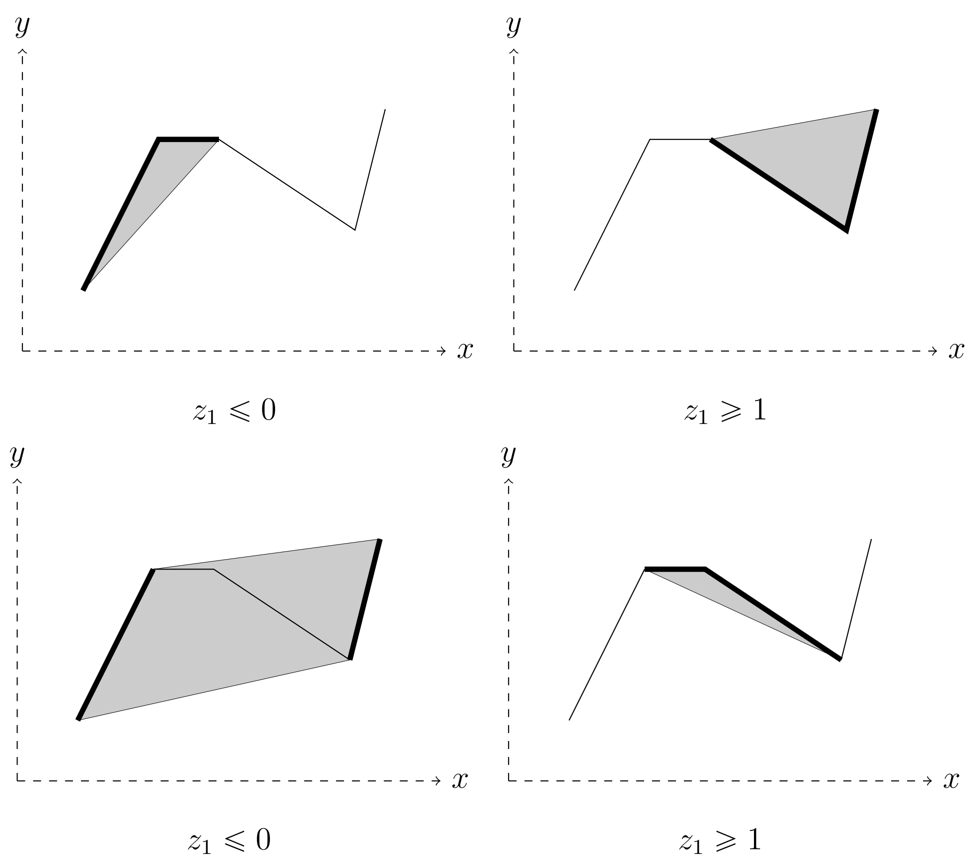


Figure 1-7: **(Top)** Good branching for one formulation of a univariate piecewise linear function, and **(Bottom)** bad, unbalanced branching from another formulation.

We attempt to formalize the quality of a formulations branching behavior through two complementary notions.

Definition 3. *Take a formulation* $F = \{ (x, w, z) \in R \mid z \in \mathbb{Z}^r \}$ *for* D *given by the LP relaxation* R*. Given* $k \in [\![r]\!]$ *and* $d \in \mathbb{Z}$*, take*

$$R_{\downarrow} = \{ (x, w, z) \in R \mid z_k \leqslant d \}$$
$$R_{\uparrow} = \{ (x, w, z) \in R \mid z_k \geqslant d + 1 \}$$

as the relaxations after down-branching *on* $z_k \leqslant d$ *and* up-branching *on* $z_k \geqslant d + 1$*, respectively. Furthermore, take*

$$D_{\downarrow} = \{ x \in D \mid \exists w, z \text{ s.t. } (x, w, z) \in F,\ z_k \leqslant d \}$$
$$D_{\uparrow} = \{ x \in D \mid \exists w, z \text{ s.t. } (x, w, z) \in F,\ z_k \geqslant d + 1 \}$$

as the portion of D *feasible after down-branching and up-branching, respectively.*

- *The formulation is* hereditarily sharp *[72, 73] if, for each* $k \in [\![r]\!]$ *and each* $d \in \mathbb{Z}$*,* $\{ (x, w, z) \in R_{\downarrow} \mid z \in \mathbb{Z}^r \}$ *and* $\{ (x, w, z) \in R_{\uparrow} \mid z \in \mathbb{Z}^r \}$ *are sharp formulations for* $D_{\downarrow}$ *and* $D_{\uparrow}$*, respectively.*

- *The formulation has* incremental branching *if, for each* $k \in [\![r]\!]$ *and each* $d \in \mathbb{Z}$*,*

$$\text{int}(\text{Conv}(D_{\downarrow})) \cap \text{int}(\text{Conv}(D_{\uparrow})) = \varnothing.$$

In words, a formulation is hereditarily sharp if it retains its sharpness after branching. Additionally, a formulation has incremental branching if branching results in subproblems that have disjoint feasible regions; this aligns with folklore wisdom in the MIP literature [23]. Indeed, we can see this borne out in Figure 1-7: both formulations are hereditarily sharp, but the first also has incremental branching, which leads to much more balanced subproblems.

These three measures of quality–strength, size, and branching behavior–are often in conflict. For example, the tools we develop in Chapter 3 show that we can always produce strong formulations with few integer variables. However, if we are not careful, these formulations can easily require exponentially many inequality constraints (cf. [130]). On the other hand, there exist structures for which you can construct formulations that are very small, with only a constant number of general inequality constraints, although in order to do this you must sacrifice formulation strength (this result appears in a preprint version of [129]). Furthermore, as we see in Chapter 4, the logarithmic formulations for univariate piecewise linear functions of Vielma and Nemhauser [135] are small and strong, but exhibit degenerate branching behavior. In this thesis we will endeavor to build formulations that balance all three metrics at once.

1.5 Existing approaches

We are now prepared to present a number of MIP formulation techniques from the literature that can be applied to any combinatorial disjunctive constraint. These standard formulations will provide a benchmark for comparing against our new formulations we will develop in this thesis.

A standard formulation for $\mathrm{CDC}(\mathcal{T})$ adapted from Jeroslow and Lowe [72] is

$$\begin{aligned}
\lambda_v &= \sum_{T \in \mathcal{T} : v \in T} \gamma_v^T \qquad \forall v \in V && \text{(1.15a)}\\
z_T &= \sum_{v \in T} \gamma_v^T \qquad \forall T \in \mathcal{T} && \text{(1.15b)}\\
\sum_{T \in \mathcal{T}} z_T &= 1 && \text{(1.15c)}\\
\gamma^T &\in \Delta^T \qquad \forall T \in \mathcal{T} && \text{(1.15d)}\\
(\lambda, z) &\in \Delta^V \times \{0,1\}^{\mathcal{T}}. && \text{(1.15e)}
\end{aligned}$$

We will call this the "multiple choice" (`MC`) formulation. This formulation has $|\mathcal{T}|$ binary variables, $\sum_{T \in \mathcal{T}} |T|$ auxiliary continuous variables, and no general inequality constraints. Additionally, it is ideal.

Using Proposition 9.3 from [130], we can construct an ideal MIP formulation with fewer binary variables:

$$\lambda_v = \sum_{T \in \mathcal{T} : v \in T} \gamma_v^T \qquad \forall v \in V \tag{1.16a}$$

$$\sum_{T \in \mathcal{T}} \sum_{v \in T} \gamma_v^T = 1 \tag{1.16b}$$

$$\sum_{T \in \mathcal{T}} \sum_{v \in T} h^T \gamma_v^T = z \tag{1.16c}$$

$$\gamma^T \geqslant 0 \qquad \forall T \in \mathcal{T} \tag{1.16d}$$

$$z \in \{0,1\}^r, \tag{1.16e}$$

where $\{h^T\}_{T \in \mathcal{T}} \subseteq \{0,1\}^r$ is some set of distinct binary vectors. This formulation is actually a generalization of (1.15), which we recover if we take $h^T = \mathbf{e}^T \in \mathbb{R}^{\mathcal{T}}$ as the canonical unit vectors. If instead we take r to be as small as possible (while ensuring that the vectors $\{h^T\}_{T \in \mathcal{T}}$ are distinct), we recover $r = \lceil \log_2(d) \rceil$. This is a *disaggregated logarithmic* (`DLog`) formulation, and is an ideal extended formulation for (1.5) with $\lceil \log_2(d) \rceil$ binary variables, $\sum_{T \in \mathcal{T}} |T|$ auxiliary continuous variables, and no general inequality constraints. The following result shows that, in the standard binary MIP setting, this is the smallest number of integer variables we may hope for.

Proposition 1. *If the sets $\mathcal{T}$ are irredundant, then any binary MIP formulation for* $\mathrm{CDC}(\mathcal{T})$ *must have at least* $\lceil \log_2(d) \rceil$ *binary variables.*

Proof. Follows as a special case of Proposition 12. □

The formulations (1.15) and (1.16) are both extended formulations that work by formulating each alternative separately and then aggregating them, rather than working directly with the combinatorial structure underlying the shared extreme points. Therefore, each of these formulations requires a copy of the multiplier γ_v^T for each set $T \in \mathcal{T}$ for which $v \in T$, and so $\sum_{i=1}^{d} |\mathcal{T}|$ auxiliary continuous variables total.

In contrast, we can construct non-extended formulations for $\mathrm{CDC}(\mathcal{T})$ that work directly on the λ variables and the underlying combinatorial structure of $\mathcal{T}$. An example of a non-extended formulation for CDC is the widely used ad-hoc formulation (see [130, Section 6] and the references therein) given by

$$\lambda_v \leqslant \sum_{T \in \mathcal{T}: v \in T} z_T \qquad \forall v \in V \tag{1.17a}$$

$$\sum_{T \in \mathcal{T}} z_T = 1 \tag{1.17b}$$

$$(\lambda, z) \in \Delta^V \times \{0,1\}^{\mathcal{T}}. \tag{1.17c}$$

We will call this formulation the "convex combination" (CC) formulation. This formulation is not necessarily ideal, and it requires no auxiliary continuous variables, d binary variables, and $|V|$ general inequality constraints.

In summary, we have seen an ideal extended formulation (1.16) for $\mathrm{CDC}(\mathcal{T})$ with relatively few binary variables, but relatively many auxiliary continuous variables. On the other end of the spectrum, we have a non-extended formulation (1.17) with no auxiliary continuous variables, but which requires relatively many binary variables and which may fail to be ideal. However, there exist special cases we can construct ideal, non-extended formulations with only $\mathcal{O}(\log(d))$ binary variables and constraints (e.g. SOS1, SOS2, and particular 2-dimensional grid triangulations [135, 133]). Moreover, these "logarithmic" formulations have proven computational efficacy for a wide swath of problem instances [131, 135]. For the remainder of this thesis, we will present systematic ways to build small, strong MIP formulations in this vein for a wide range of disjunctive constraints, including those introduced in Chapter 1.3.

1.6 Contributions of this thesis

The contributions of this thesis are as follows.

Chapter 2 We study in detail the independent branching framework of Vielma and Nemhauser [133], a combinatorial way to build MIP formulations for disjunctive constraints. We provide an exact characterization of the expressive power of this framework by providing a graphical answer to the question: Does any independent branching formulation exist for a given constraint? We answer in the affirmative for many constraints of interest (piecewise linear functions, SOSk constraints, discretizations of multilinear functions, obstacle avoidance), but negatively for others (cardinality constraints). Provided that *any* independent branching formulation exists, we show that *each* independent branching formulation corresponds to a biclique cover for a particular graph. In other words, the question of formulating a disjunctive constraint reduces to a purely combinatorial problem. Using this insight, we are able to systematically construct small independent branching formulations for many of our motivating problems, and provide bounds on how far these heuristic constructions are from the smallest possible.

Chapter 3 We study a geometric way to construct MIP formulations through what is known as the embedding approach. We provide a way to construct ideal formulations for any combinatorial disjunctive constraint that is purely geometric, its complexity hinging solely on the computation of spanning hyperplanes for a set of points. The embedding approach very naturally allows us to consider MIP formulations that use general integer (as opposed to binary) integer variables, and we provide examples where this extra freedom allows us to create much smaller formulations than would be possible otherwise. However, we show that this extra freedom does not afford any improvement in formulation size for combinatorial disjunctive constraints. Nevertheless, we are able to produce new MIP formulations for univariate piecewise linear functions and the annulus that are strong (ideal), and small (logarithmic in d). Moreover, we are able to design them to use general integer variables in such a way that induces desirable branching behavior.

Chapter 4 We perform an in-depth computational study of MIP formulations for piecewise linear functions. We show that the new formulations derived in Chapters 2 and 3 afford a substantial computational improvement over myriad existing approaches:

- For univariate functions: A 1.5-3x speed-up on harder instances.

- For bivariate functions: Over an order-of-magnitude speed-up.

We also showcase computational modeling tools developed in tandem, which offer a high-level interface for writing and solving optimization models containing piecewise linear functions. We close the chapter by turning to high-dimensional piecewise linear functions arising in the context of deep learning, and develop a new ideal formulation and valid inequalities for these structures.

Chapter 5 We close this thesis with a more speculative question: Given an arbitrary disjunctive constraint, what do we have to sacrifice in order to construct MIP formulations with very few integer variables? We show that, if we allow ourselves to leave spurious "integer holes" in the interior of our MIP formulations, we can produce ideal formulations for any disjunctive constraint with only two integer variables and a linear number of linear inequality constraints. Additionally, we can reduce the number of constraints to a small constant number ($\leqslant 6$) for specific structures like univariate piecewise linear functions and the annulus. We provide a framework to encompass these *MIP-with-holes formulations*, and show how they fit together with standard MIP algorithmic approaches (e.g. branch-and-bound and cutting planes).

1.7 Notation

In Table 1.1 we provide a summary of some of the notation we will return to throughout the course of this thesis.

Notation	Definition	Description
$\mathrm{Conv}(S)$	-	Convex hull of set S
$[\![d]\!]$	$\{1, \ldots, d\}$	All integers from 1 to d
$[\![c, d]\!]$	$\{c, c+1, \ldots, d\}$	All integers from c to d
$\mathbb{R}^n_{\geqslant 0}$	$\{\, x \in \mathbb{R}^n \mid x \geqslant 0 \,\}$	Nonnegative orthant in n-dimensional space
$\mathrm{Proj}_x(R)$	$\{\, x \mid \exists y, z \text{ s.t. } (x, y, z) \in R \,\}$	The projection of set R onto x variables
Δ^V	$\left\{\, \lambda \in \mathbb{R}^V_{\geqslant 0} \,\middle\vert\, \sum_{v \in V} \lambda_v = 1 \,\right\}$	Standard simplex on ground set V
$\mathrm{supp}(\lambda)$	$\{\, v \in V \mid \lambda_v \neq 0 \,\}$	Nonzero components (*support*) of λ
$P(T)$	$\left\{\, \lambda \in \Delta^V \,\middle\vert\, \mathrm{supp}(\lambda) \subseteq T \,\right\}$	Face of standard simplex given by components T
$\mathrm{CDC}(\mathcal{T})$	$\bigcup_{i=1}^d P(T^i)$	Combinatorial disjunctive constraint for $\mathcal{T} = (T^i)_{i=1}^d$
$\mathrm{ext}(P)$	-	Extreme points of polyhedra P
$\mathrm{gr}(f)$	$\{\, (x, f(x)) \mid x \in \Omega \,\}$	Graph of the function f over the domain Ω
$\mathrm{int}(S)$	-	The interior of set S
$r(H)$	$\max_{E \in \mathcal{E}} \lvert E \rvert$	Rank of the hypergraph $H = (V, \mathcal{E})$
$[V]^2$	$\{\, \{u, v\} \in V \times V \mid u \neq v \,\}$	All unordered pairs of elements in V
$\mathrm{Em}(\mathcal{P}, H)$	$\bigcup_{i=1}^d P^i \times \{h^i\}$	Embedding of $\mathcal{P} = (P^i)_{i=1}^d$ with $H = (h^i)_{i=1}^d)$
$Q(\mathcal{P}, H)$	$\mathrm{Conv}(\mathrm{Em}(\mathcal{P}, H))$	Convex hull of embedding
$\mathrm{Slice}(R; z)$	$\{\, x \mid \exists w \text{ s.t. } (x, w, z) \in R \,\}$	The slice of of set R when z variables are fixed
$Z(R)$	$\{\, z \in \mathbb{Z}^r \mid \exists x, w \text{ s.t.} (x, w, z) \in R \,\}$	Feasible integer values for formulation R
$\mathrm{aff}(H)$	-	Affine hull of H
$L(H)$	$\{\, y - h^1 \mid y \in \mathrm{aff}(H) \,\}$	Linear space parallel to $\mathrm{aff}(H)$ (where $h^1 \in H$)
$M(b)$	$\{\, y \in L(H) \mid b \cdot y = 0 \,\}$	The hyperplane in $L(H)$ normal to b
$\mathrm{Vol}(D)$	-	Volume of set D
$A * B$	$\{\, \{u, v\} \mid u \in A, v \in B \,\}$	Unordered pairs of elements in A and B

Table 1.1: Notation used throughout the paper.

Chapter 2

A combinatorial way to construct formulations.

One relatively generic and versatile way to build small, strong formulations is the *independent branching* (IB) scheme framework introduced by Vielma and Nemhauser [135]. The approach is to find some (particularly structured) polyhedra $Q^{1,j}$ and $Q^{2,j}$ such that the disjunctive set in (1.5) can be rewritten as

$$D \equiv \bigcup_{i=1}^{d} P^i = \bigcap_{j=1}^{t} \left(Q^{1,j} \cup Q^{2,j} \right). \tag{2.1}$$

This represents the disjunctive constraint in term of a series of simple choices between two alternatives. Given such a representation, it is often straightforward to construct a simple, small, and ideal formulation for (1.5) by formulating each of the t alternatives separately, and then combining them. In particular, when the polyhedra P^i are $\mathcal{V}$-polyhedra, the construction of the independent branching scheme-based formulation is purely combinatorial. We can therefore approach formulating (1.5) combinatorially, by studying the shared structure amongst the extreme points. In this chapter we generalize and provide a systematic study of the applicability and limitations of the independent branching approach for combinatorial disjunctive constraints.

2.1 Independent branching schemes

The independent branching approach is a logically equivalent way of expressing a CDC in terms of a conjunction of dichotomies: that is, as a series of choices between two simple options. This approach is parsimonious: if you are given an independent branching scheme for a particular CDC, it is straightforward to construct an ideal formulation whose size is on the order of the number of dichotomies. For our purposes, we present a generalized notion, where we allow potentially more than two alternatives.

Definition 4. *A* k-way independent branching scheme *for* CDC($\mathcal{T}$) *is given by a family of sets* $(L_1^j, \ldots, L_k^j)$ *(where each* $L_i^j \subseteq V$*) for* $j \in [\![t]\!]$*, where*

$$\text{CDC}(\mathcal{T}) = \bigcap_{j=1}^{t} \left(\bigcup_{i=1}^{k} P(L_i^j) \right). \tag{2.2}$$

We say that such an IB scheme has *depth* t, and that each $j \in [\![t]\!]$ yields a corresponding *level* of the IB scheme, $\bigcup_{i=1}^k P(L_i^j)$, given by the k *alternatives* $\{P(L_i^j)\}_{i=1}^k$.

In this form, we have replaced the monolithic disjunctive constraint CDC($\mathcal{T}$) with t disjunctive constraints, each of which require the selection between k alternatives. The hope here is that $t, k \ll d$, as we can then use standard techniques to construct a corresponding small MIP formulation for the independent branching scheme directly.

Proposition 2. *Given* $\mathcal{T} = (T^i \subseteq V)_{i=1}^d$ *and an independent branching scheme* $\{(L_1^j, \ldots, L_k^j)\}_{j=1}^t$ *for* CDC($\mathcal{T}$)*, the following is a valid MIP formulation for* CDC($\mathcal{T}$)*:*

$$\sum_{v \notin L_i^j} \lambda_v \leqslant 1 - z_i^j \quad \forall j \in [\![t]\!], \forall i \in [\![k]\!] \tag{2.3a}$$

$$\sum_{i=1}^{k} z_i^j = 1 \quad \forall j \in [\![t]\!] \tag{2.3b}$$

$$\lambda \in \Delta^V \tag{2.3c}$$

$$z^j \in \{0,1\}^k \quad \forall j \in [\![t]\!]. \tag{2.3d}$$

The formulation is known to be ideal for $k = 2$ [133, 135]. It has no auxiliary continuous variables, kt auxiliary binary variables, and kt general inequality constraints.

An equivalent way of understanding independent branching representations, which we will be using for the remainder of this work, is by eschewing the polyhedra $P(L_i^j)$ and working directly on the underlying set L_i^j. That is, a valid k-way IB scheme satisfies the condition that

$$T \subseteq V \text{ is a feasible set} \iff \forall j \in [\![t]\!],\ \exists i \in [\![k]\!] \text{ s.t. } T \subseteq L_i^j.$$

First, we observe that, due to our assumption that $\mathcal{T}$ covers the ground set, we have that for each element $v \in V$ and level $j \in [\![t]\!]$, there will be at least one alternative $i \in [\![k]\!]$ such that $v \in L_i^j$. We will use this extensively in the analysis to come, as it simplifies some otherwise tedious case analyses. Second, we see that this definition can capture potential schemes with a variable number of alternatives in each level by adding empty alternatives $L_i^j = \emptyset$, provided we take k as the maximum number of alternatives for all levels. For notational simplicity, we say that a 2-way IB scheme is a *pairwise IB scheme*, and in this case we write the sets as $\{(L^j, R^j)\}_{j=1}^t$ as in [135]. In contrast, we will call the case with $k > 2$ a *multi-way IB scheme*.

2.2 What does *independent branching* mean?

2.2.1 Constraint branching via independent branching schemes

The standard algorithm used to solve mixed-integer programming problems is some variation of branch-and-bound [86], which implicitly enumerates all possible values for the integer variables. In its simplest form applied to a binary MIP formulation, a sequence of problems are solved: the optimization problem over the relaxation of the MIP formulation is solved, after which a binary variable z_i is chosen for *branching*. That is, the current problem is split into two subproblems: one with the additional constraint $z_i \leqslant 0$, another with $z_i \geqslant 1$. Repeating this procedure, the subproblems form a (binary) tree whose leaves correspond to all 2^r possible values for r binary variables. At any given subproblem, the augmented relaxation to be optimized over is described by the set of binary variables fixed to zero, and the set of those fixed to one.

The spirit of constraint branching is to allow richer branching decisions. For example, a branching decision might be between k alternatives of the form $\{Q^i\}_{i=1}^k$, where each Q^i is formed by adding a general inequality constraint to the existing relaxation at the current node. This concept has significant overlap with the broader field of constraint programming [8, 70], which has been recognized and exploited in the mixed-integer programming literature [1, 7, 65, 110, 118]. More complex constraint branching can often lead to a more balanced branch-and-bound tree, which in turn can significantly improve computational performance (see, for example, [130, Section 8] and [139] for more discussion). Combinatorial disjunctive constraints are a natural setting to apply constraint branching directly on the continuous λ variables [13, 43, 75, 76, 100]. Indeed, the classical examples of the SOS1 and SOS2 constraints [13] show that we do not necessarily require a MIP formulation (or the auxiliary binary variables z) for modeling combinatorial disjunctive constraints, as the disjunction can be enforced directly through constraint branching on the λ variables. These constraint branching approaches without auxiliary binary variables can be implemented in an ad-hoc branch-and-bound procedure, or through branching callbacks available in some MIP solvers, such as CPLEX. In theory, this approach should outperform a MIP formulation like (1.17) that introduces additional variables and constraints. However, realizing this performance advantage in practice can require significant effort and technical expertise. For instance, Vielma et al. [133, 134] observe that the basic formulation (1.17) clearly outperformed the SOS2 branching implementation in CPLEX v9.1. However, CPLEX v11 implemented an optimized version of SOS2 branching that used the advanced branch selection techniques available for variable branching, reversing this performance gap with respect the MIP formulation approach.

One way to avoid re-implementing the advanced branching selection techniques for a new constraint branching approach is by constructing a MIP formulation that automatically inherits the advanced constraint branch selection, but using the solvers traditional variable branching [7, 135]. Consider a constraint branching approach that has t branching options, each of which creates k branches, and that each constraint added has support on the λ variables, with variable coefficients in $\{0, 1\}$ and a zero right-hand-side. That is, branch $i \in \llbracket k \rrbracket$ of branching option $j \in \llbracket t \rrbracket$ adds a constraint of the form $\sum_{v \notin L_i^j} \lambda_v \leqslant 0$. This is equivalent to a *multi-variable* branching approach that fixes groups of variables to zero, and it includes as special cases most constraint branching approaches, including SOS1 and SOS2 branching. Then (2.3) is a MIP formulation for this multi-variable constraint branching scheme, as variable branching on z_i^j enforces constraint branching option j of branch i on the λ variables.

This connection highlights the natural theoretical equivalence between a multi-variable branching and an independent branching formulation. A practical difference between the two is that direct multi-variable branching must implement an explicit branch selection and implementation routine, while an independent branching formulation inherits the variable branching selection and implementation routines of the MIP solver. The upshot of this is that the independent branching formulation must provide a complete catalog of all possible branching options up-front (i.e. through the formulation), while direct multi-variable branching can have a large catalog of branching options that are implicitly defined by the branching routines.

2.2.2 Independence in formulation-induced branching schemes

As discussed in [135, Section 3], the connection between MIP formulations and branching schemes for CDCs can be used to explain in what sense an independent branching scheme is "independent." In the previous subsection, we saw that the branching scheme on λ induced by formulation (2.3) is precisely the multi-variable branching associated to the corresponding independent branching scheme. In contrast, formulations that are not based on IB schemes (e.g. (1.16)) do not necessarily induce a multi-variable branching. However, we can sometimes interpret the induced effect on the λ variables as a multi-way branching scheme that fixes the λ variables in a *non-independent* way.

For example, consider the SOS2(5) constraint, given by the family of sets $\mathcal{T}_5^{\text{SOS2}} = (\{1,2\},\{2,3\},\{3,4\},\{4,5\})$. For this particular instance and for $(h^T)_{T\in\mathcal{T}}$ given by $h^{\{1,2\}} = (1,1)$, $h^{\{2,3\}} = (1,0)$, $h^{\{3,4\}} = (0,1)$, and $h^{\{4,5\}} = (0,0)$, formulation (1.16) is

$$\lambda_1 = \gamma_1^{\{1,2\}}, \quad \lambda_2 = \gamma_2^{\{1,2\}} + \gamma_2^{\{2,3\}}, \lambda_3 = \gamma_3^{\{2,3\}} + \gamma_3^{\{3,4\}}, \tag{2.4a}$$

$$\lambda_4 = \gamma_4^{\{3,4\}} + \gamma_4^{\{4,5\}}, \quad \lambda_5 = \gamma_5^{\{4,5\}} \tag{2.4b}$$

$$\gamma_1^{\{1,2\}} + \gamma_2^{\{1,2\}} + \gamma_2^{\{2,3\}} + \gamma_3^{\{2,3\}} + \gamma_3^{\{3,4\}} + \gamma_4^{\{3,4\}} + \gamma_4^{\{4,5\}} + \gamma_5^{\{4,5\}} = 1 \tag{2.4c}$$

$$\gamma_1^{\{1,2\}} + \gamma_2^{\{1,2\}} + \gamma_2^{\{2,3\}} + \gamma_3^{\{2,3\}} = z_1 \tag{2.4d}$$

$$\gamma_1^{\{1,2\}} + \gamma_2^{\{1,2\}} + \gamma_3^{\{3,4\}} + \gamma_4^{\{3,4\}} = z_2 \tag{2.4e}$$

$$\gamma_v^S \geqslant 0 \quad \forall v, T \tag{2.4f}$$

$$(\lambda, z) \in \Delta^5 \times \{0,1\}^2 \tag{2.4g}$$

and a pairwise IB formulation (simplified slightly from (2.3)) is

$$\lambda_1 + \lambda_2 \leqslant z_1, \qquad \lambda_4 + \lambda_5 \leqslant 1 - z_1 \tag{2.5a}$$

$$\lambda_3 \leqslant z_2, \qquad \lambda_1 + \lambda_5 \leqslant 1 - z_2 \tag{2.5b}$$

$$(\lambda, z) \in \Delta^5 \times \{0,1\}^2. \tag{2.5c}$$

In Figure 2-1, we see the first two levels of the branch-and-bound trees for both formulations for the cases where we choose either z_1 or z_2 for which to branch on first. We observe that, for formulation (2.4), the variables λ_v that a given branching decision is able to prove are zero depends on the previous branching decisions in the branch-and-bound tree, while this is not the case for the independent branching formulation (2.5). For example, with (2.4) if we first branch down on z_2 (i.e. $z_2 \leqslant 0$), we are able to prove that $\lambda_1 = 0$. If we choose instead to first branch down on z_1 (i.e. $z_1 \leqslant 0$), we are able to prove that $\lambda_1 = \lambda_2 = 0$. However, if we branch down on z_2 and then branch down on z_1, we prove that $\lambda_1 = \lambda_2 = 0$, but we are also able to prove that $\lambda_3 = 0$, which we could not prove without the combination of the two branching decisions. Indeed, we see that regardless of the branching decision we make, we will not be able to prove that $\lambda_3 = 0$ until the second level of the branching tree with formulation (2.4).

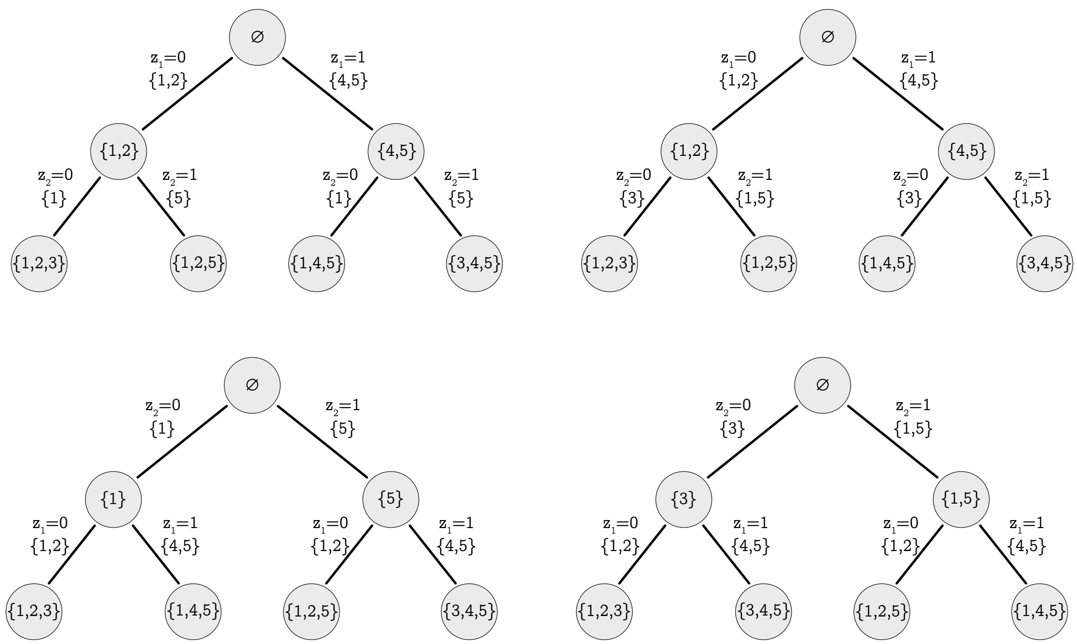


Figure 2-1: The branch-and-bound trees for **(Left)** (2.4) and **(Right)** (2.5), when **(Top row)** z_1 is first to branch on, and then z_2, and when **(Bottom row)** z_2 is first to branch on, and then z_1. Inside each node is the set $I \subset [\![5]\!]$ of all components v for which the algorithm has been able prove that $\lambda_v = 0$ at this point in the algorithm via branching decisions. The text on the lines show the current branching decision (e.g. $z_2 \geqslant 1$), and the set of components $v \in [\![5]\!]$ for which the (a) subproblem is able to prove that $\lambda_v = 0$ *independently* of any other branching decisions (e.g. $z_2 \geqslant 1$ is the only additional branching constraint added to the original relaxation). This figure is adapted from [135, Figure 2].

In contrast, each branching decision with the independent branching formulation (2.5) is able to fix components of λ to zero, *independent* of the location in the tree and of the previous branching decisions. For example, branching down or up on z_2 is always able to prove either $\lambda_3 = 0$ or $\lambda_1 = \lambda_5 = 0$, independently. Consequentially, for every component $v \in V$, there exists a branching decision that is able to prove that $\lambda_v = 0$ at the first level of the branch-and-bound tree, which is not the case with formulation (2.4) and $v = 3$, as mentioned above. Having this independence property is a restriction on the branching scheme, but has the potential to simplify branching rules (i.e. choosing which variable z_i to branch on), a notoriously difficult and computationally important part of the algorithmic performance of a MIP solver (see, for example, [2]). Furthermore, we see that independent branching rules guarantee that the solver can prove any component of λ is zero at the very beginning of the tree.

Finally, we note that MIP formulations that are not independent branching formulations can still exhibit the independent branching behavior. For example, if we had selected the encoding $(h^T)_{T \in \mathcal{T}}$ to be given by given by $h^{\{1,2\}} = (1,1)$, $h^{\{2,3\}} = (1,0)$, $h^{\{3,4\}} = (0,0)$, and $h^{\{4,5\}} = (0,1)$, formulation (1.16) would satisfy the independent branching property. However, independent branching formulations provide an immediate proof that the property holds, which is not the case for general MIP formulations.

2.3 Independent branching scheme representability

We now offer a complete characterization of the expressive power of the independent branching formulation framework. To start, we observe that, given $\mathcal{T}$, it is not sufficiently general to capture every possible formulation for CDC($\mathcal{T}$). For example, there is the restriction that each alternative $P(L_i^j)$ restricts the λ variables to lie on a single face of the standard simplex. A natural first question is then: given a family of sets $\mathcal{T}$, do any k-way IB schemes exist for CDC($\mathcal{T}$)? We provide an answer, based on a graphical characterization of the constraint.

2.3.1 A graphical characterization

Definition 5. *Let $H \stackrel{\text{def}}{=} (V, \mathcal{E})$ be a hypergraph with hyperedge set $\mathcal{E} \subseteq 2^V$.*

- *The* rank *of H is $r(H) \stackrel{\text{def}}{=} \max_{E \in \mathcal{E}} |E|$.*

- *A* (weakly) independent set *of H is a set $U \subseteq V$ that does not contain any element of $\mathcal{E}$ as a subset.*

- *The* conflict hypergraph *of* $\mathcal{T}$ *is* $H^c_{\mathcal{T}} \overset{\text{def}}{=} (V, \mathcal{E}_{\mathcal{T}})$*, where*

$$\mathcal{E}_{\mathcal{T}} \overset{\text{def}}{=} \{\, E \subseteq V \mid \; E \textit{ is a minimal infeasible set} \,\}\,.$$

Lemma 1. *The maximal independent sets T in $H^c_{\mathcal{T}}$ are exactly the sets $T \in \mathcal{T}$.*

Proof. If $T \in \mathcal{T}$, it is obviously a feasible set, and so we have immediately that T is an independent set in $H^c_{\mathcal{T}}$. If it is not maximal, then we could add some $v \in V \backslash T$ and maintain independence. However, any independent set of $H^c_{\mathcal{T}}$ must be contained in a feasible set $T' \in \mathcal{T}$, which would violate our irredundancy assumption (i.e. $T \cup \{v\} \subseteq T' \in \mathcal{T}$, $T \in \mathcal{T}$, and $T \subsetneq T \cup \{v\} \subseteq T'$).

If T is a maximal independent set in $H^c_{\mathcal{T}}$, then it must be a feasible set with respect to $\mathcal{T}$ as well. As it is maximal, there is no set $\hat{T} \in \mathcal{T}$ with $T \subsetneq \hat{T}$, and so we must have $T \in \mathcal{T}$ as well. $\square$

Theorem 1. *A k-way IB scheme for* $\mathrm{CDC}(\mathcal{T})$ *exists if and only if $r(H^c_{\mathcal{T}}) \leqslant k$. In particular, if $\mathcal{E}_{\mathcal{T}} = \left\{E^j = \{e^j_1, \ldots, e^j_{|E^j|}\}\right\}_{j=1}^{t}$ is the hyperedge set for the conflict hypergraph $H^c_{\mathcal{T}}$, then an $r(H^c_{\mathcal{T}})$-way IB scheme for* $\mathrm{CDC}(\mathcal{T})$ *is given by*

$$L^j_i = \begin{cases} V\backslash\{e^j_i\} & i \leqslant |E^j| \\ \varnothing & o.w. \end{cases} \qquad \forall i \in [\![r(H^c_{\mathcal{T}})]\!],\; j \in [\![t]\!]. \tag{2.6}$$

Proof. To show the "if" direction, it suffices to show the validity of (2.6). First note that every minimally infeasible set $E^j \in \mathcal{E}_{\mathcal{T}}$ is rendered infeasible by level j, which implies that every infeasible set is rendered infeasible as well. Then note that for any $T \in \mathcal{T}$ and for any $j \in [\![t]\!]$, we have $E^j \nsubseteq T$, so there exists $i \in [\![|E^j|]\!]$ such that $e^j_i \in E^j \backslash T$. Hence, $T \in L^j_i$ and T is feasible for level j.

To show the "only if" direction, assume for a contradiction that there exists a k-way IB scheme with $k \leqslant r(H^c_{\mathcal{T}}) - 1$. Take a minimal infeasible set $E = \{e_1, \ldots, e_r\} \in \mathcal{E}_{\mathcal{T}}$, where $r = r(H^c_{\mathcal{T}})$. Then take $j \in [\![t]\!]$ as a level of the IB scheme that renders E infeasible. By the minimality of E, we have that, for all $\ell \in [\![r]\!]$, there exists some $i(\ell) \in [\![k]\!]$ such that $E(\ell) \overset{\text{def}}{=} E\backslash\{e_\ell\} \subseteq L^j_{i(\ell)}$. As $k < r$, we may apply the pigeonhole principle to see that there must exist some distinct $\ell^1, \ell^2 \in [\![r]\!]$ such that $i(\ell^1) = i(\ell^2)$, and such that $E(\ell^1) \subseteq L^j_{i(\ell^1)}$ and $E(\ell^2) \subseteq L^j_{i(\ell^2)}$. As $E = E(\ell^1) \cup E(\ell^2)$ and $L^j_{i(\ell^1)} = L^j_{i(\ell^2)}$, this implies that $E \subseteq L^j_{i(\ell^1)}$, which contradicts our supposition that level j rendering E infeasible. □

Throughout, we will say that CDC($\mathcal{T}$) is k-way IB-representable (or pairwise IB-representable for $k = 2$) if it admits a k-way IB scheme.

2.3.2 Cardinality constraints

Our first application of Theorem 1 is to derive a strong restriction on the existence of multi-way IB schemes for the cardinality constraint.

Corollary 1. *The cardinality constraint of degree ℓ given by the family of sets $\mathcal{T} \equiv \mathcal{T}^{\text{card}}_{n,\ell}$ is k-way IB-representable if and only if $k > \ell$.*

Proof. Direct from Theorem 1 by observing that $r(H^c_{\mathcal{T}}) = \ell + 1$. □

We observe that the IB scheme (2.6), when applied to the cardinality constraint, is a natural MIP formulation for the "conjunctive normal form" [11], and is unlikely to be practical for even moderately large ℓ. In addition, both specialized constraint branching schemes for cardinality constraints [68] and the binary variable branching induced by standard formulations for cardinality constraints are quite imbalanced. The existence of a pairwise independent branching scheme for cardinality constraints would likely have finally produced the sought-after balanced constraint branching. However, Corollary 1 implies that such a balanced constraint branching cannot be produced via IB schemes; or, equivalently, by constraint branchings that do not use general inequality constraints (i.e. are only multi-variable branchings).

2.3.3 Polygonal partitions of the plane

Consider a (nonconvex) bounded region in the plane $\Omega \subset \mathbb{R}^2$ that describes all possible locations for a UAV, as described in Chapter 1.3.6. Assume that Ω can be partitioned into polyhedra $\{P^i\}_{i=1}^d$ such that $\bigcup_{i=1}^d P^i = \Omega$ and their interiors do not overlap

($\mathrm{int}(P^i) \cap \mathrm{int}(P^j) = \emptyset$ for each distinct $i, j \in [\![d]\!]$). We note that this partition will not, in general, be unique, and its selection can have a significant effect on questions of representability or formulation size. Figure 2-2 illustrates this for a convex region with a "hole," with three ways to partition the resulting nonconvex region into convex polyhedra. Once this partition is fixed, we describe each region P^i as a $\mathcal{V}$-polyhedra, and so the corresponding combinatorial disjunctive constraint is given by $\mathcal{T} = (\mathrm{ext}(P^i))_{i=1}^d$ and $V = \bigcup\{T \in \mathcal{T}\}$. We additionally forbid partitions with "internal vertices" by requiring that

$$v \in P^i \iff v \in \mathrm{ext}(P^i) \quad \forall i \in [\![d]\!], v \in V, \tag{2.7}$$

so that $\mathcal{T}$ corresponds to the maximal elements of a polyhedral complex [140, Section 5.1]. For example, the second and third partitions in Figure 2-2 satisfy this condition, while the first does not.

In this setting, minimal infeasible sets have a natural characterization.

Theorem 2. *Take bounded $\Omega \subset \mathbb{R}^2$ and a polyhedral partition $\{P^i\}_{i=1}^d$ of Ω satisfying the internal vertex condition (2.7). If $\mathcal{T} = \{\mathrm{ext}(P^i)\}_{i=1}^d$, then $r(H^c_{\mathcal{T}}) \leqslant 3$.*

Proof. Take some minimal infeasible hyperedge $E \in \mathcal{E}_{\mathcal{T}}$ of $H^c_{\mathcal{T}}$, assuming for contradiction that $r = |E| > 3$, and label the points $E = \{v^i\}_{i=1}^r$. First, we show that the points may not be in general position, i.e. that without loss of generality (w.l.o.g.) $v^r \in \mathrm{Conv}(\{v^i\}_{i=1}^{r-1})$. Then, we argue that the points not being in general position implies that $\{v^i\}_{i=1}^{r-1}$ is also an infeasible set, violating the minimality condition.

Assume for contradiction that the points are in general position; that is, that none can be written as a convex combination of the others. This implies that $\mathrm{ext}(\mathrm{Conv}(E)) = E$. Assume that the ordering $\{v^1, \ldots, v^r\}$ forms a path around the edges of $\mathrm{Conv}(E)$; that is, v^i and v^j both lie on an edge of $\mathrm{Conv}(E)$ if and only if $|i - j| = 1$ or $\{i, j\} = \{1, r\}$.

Choose some set $T^1 \in \mathcal{T}$ and some $2 < j < r$ such that $v^1, v^j \in T^1$ and $v^2 \notin T^1$; the associated polyhedron is P^1. Such a set exists, else E is not a minimal infeasible set (choose instead $E \backslash \{v^2\}$). Now choose $T^2 \in \mathcal{T}$ such that $v^2, v^r \in T^2$; the associated polyhedron is P^2. Such as set exists, as $\{v^2, v^r\} \subsetneq E$ and E is minimal. As the nodes v^1, v^2, v^j, v^r are interlaced along the boundary of $\text{Conv}(E)$, we have that $\text{Conv}(\{v^1, v^j\}) \cap \text{Conv}(\{v^2, v^r\}) \subset \text{Conv}(E)$ is nonempty. As each of the four points is on the boundary of $\text{Conv}(E)$, and the points are in general position, it follows that $\text{Conv}(\{v^1, v^j\}) \cap \text{Conv}(\{v^2, v^r\}) = \text{int}(\text{Conv}(\{v^1, v^j\})) \cap \text{int}(\text{Conv}(\{v^2, v^r\}))$. Therefore, there must exist some point y with $y \in \text{int}(\text{Conv}(\{v^1, v^j\})) \subseteq \text{int}(P^1)$ and $y \in \text{int}(\text{Conv}(\{v^2, v^r\})) \subseteq \text{int}(P^2)$. However, this implies that $\text{int}(P^1) \cap \text{int}(P^2) \neq \emptyset$, which contradicts the assumption that our sets partition the region Ω.

Finally, it just remains to show that $\{v^i\}_{i=1}^{r-1}$ is also an infeasible set, and therefore $\{v^i\}_{i=1}^{r}$ cannot be a minimal infeasible set. Assume for contradiction that it is not: i.e. that there exists some j such that $\{v^i\}_{i=1}^{r-1} \subseteq \text{ext}(P^j)$. But this implies that $v^r \in V$ and $v^r \in \text{Conv}(E) \subseteq P^j$, yet $v^r \notin \text{ext}(P^j)$, a contradiction of the internal vertices assumption. $\square$

In other words, every polyhedral partition of the plane is 3-way independent branching-representable, and pairwise IB representability can be checked in time polynomial in $|\mathcal{T}|$ (for example, by enumerating the subsets of V of cardinality 3). To illustrate, in Figure 2-2 we depict the three possible cases for a partition with respect to Theorem 2: 1) it does not satisfy the internal vertices condition, 2) it admits a pairwise IB scheme ($r(H^c_{\mathcal{T}}) = 2$), or 3) it does not admit a pairwise IB scheme, but does admit a 3-way IB scheme ($r(H^c_{\mathcal{T}}) = 3$).

Furthermore, we argue that we can always represent a obstacle avoidance constraint in such a way that it admits a pairwise IB scheme. Inspecting Figure 2-2, we see that the region Ω is the same in each, and it is only the partition of Ω that can potentially lead to constraints that are not pairwise IB-representable. Therefore, the obstacle avoidance constraint is invariant to the specification of the partition, and if any polyhedral partitioning exists, then it is always possible to construct one that satisfies the conditions of Theorem 2.[1] We provide a sketch of the argument and construction techniques in Appendix B.

[1]Note that this result does not carry over to piecewise linear functions defined over a partition of Ω, as the choice of the partition is intimately connected with the values the function may take.

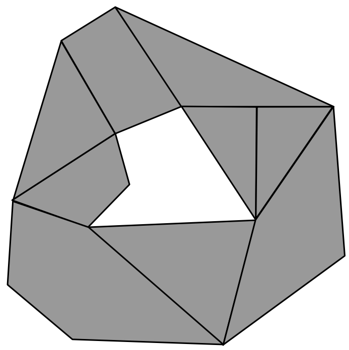


Figure 2-2: Partitions of a nonconvex region in the plane obtained by removing a central non-convex portion from a convex polyhedron. **(Left)** The first partition does not satisfy the internal vertices condition (2.7), **(Center)** the second partition admits a pairwise IB scheme, and **(Right)** the third partition admits a 3-way IB scheme but not a pairwise one.

2.3.4 Pairwise IB-representable constraints

We now return to the constraints introduced in Chapter 1.3 which always admit pairwise independent branching schemes.

The SOS2 constraint

Our first example of a constraint that is always pairwise IB-representable is the SOS2(N) constraint. Recall that $N = d+1$, $V = \llbracket N \rrbracket$, and $\mathcal{T} \equiv \mathcal{T}_d^{\text{SOS2}} = (\{\tau, \tau+1\})_{\tau=1}^d$ for SOS2. Then $\mathcal{E}_{\mathcal{T}} = \{ \{\tau, \tau+t\} \mid \tau, \tau+t \in \llbracket N \rrbracket,\ t \geqslant 2 \}$, $r(H_{\mathcal{T}}^c) = 2$, and formulation (2.6) has depth $t = \binom{N}{2} - N + 1$. However, Vielma and Nemhauser [135] construct a pairwise IB scheme for SOS2(N) constraints with depth logarithmic in N. The construction is built around a Gray code [120], or sequence of distinct binary vectors $\{h^i\}_{i=1}^d \subseteq \{0,1\}^{\lceil \log_2(d) \rceil}$ where each adjacent pair (h^i, h^{i+1}) differs in exactly one component. Notationally, here and throughout, take $h^0 \stackrel{\text{def}}{=} h^1$ and $h^{d+1} \stackrel{\text{def}}{=} h^d$. The pairwise IB scheme is then given by

$$L^j = \left\{ \tau \in \llbracket N \rrbracket \;\middle|\; h_j^{\tau-1} = 1 \text{ or } h_j^{\tau} = 1 \right\}$$
$$R^j = \left\{ \tau \in \llbracket N \rrbracket \;\middle|\; h_j^{\tau-1} = 0 \text{ or } h_j^{\tau} = 0 \right\}$$

for each $j \in [\![\lceil \log_2(d) \rceil]\!]$. We observe that the resulting formulation matches the lower bound from Proposition 1 with respect to the number of binary variables and is significantly smaller than formulation (2.6).

SOSk

In this case where $\mathcal{T} \equiv \mathcal{T}^{\text{SOS}}_{N,k}$, then $\mathcal{E}_{\mathcal{T}} = \{\, \{\tau, \tau+t\} \mid \tau, \tau+t \in [\![N]\!],\ t \geqslant k+1 \,\}$ and $r(H^c_{\mathcal{T}}) = 2$.

Discretization of multilinear functions

Each infeasible set $U \in \mathcal{E}_{\mathcal{T}}$ must necessarily contain two elements $v, w \in U$ with $||v - w||_\infty > 1$, and so we have that U can be reduced to the infeasible pair $\{v, w\}$. Therefore, $r(H^c_{\mathcal{T}}) = 2$, and any discretization of this form is pairwise IB-representable.

Grid triangulations

We show that $r(H^c_{\mathcal{T}}) = 2$ by seeing that for any infeasible set $U \subseteq V$ there exist some distinct $v, w \in U$ such that $\{v, w\}$ is infeasible. Analogously to the case with discretizations of multilinear functions above, if there are some $v, w \in U$ such that $||v - w||_\infty > 1$, then there does not exist any triangle on the grid that contains both, so $\{v, w\}$ is also an infeasible set. Otherwise, we have that $U \subset \{r, r+1\} \times \{s, s+1\}$ for some r, s, and that U contains elements in both of the triangles in this square. For each of the two triangles, we can select an element of U that is not contained in the other triangle, which yields an infeasible pair contained in U. Therefore, any grid triangulation is pairwise IB-representable.

Higher-dimensional grid triangulations

Take $\mathcal{T}$ as corresponding to the η-dimensional standard grid triangulation, as given in (1.13). By the same argument as above for discretizations of multilinear functions, we can restrict ourselves to infeasible sets $U \in \mathcal{E}_{\mathcal{T}}$ that are contained completely in one subrectangle; w.l.o.g., take $U \subseteq \{0, 1\}^\eta$. We show that $r(H^c_{\mathcal{T}}) = 2$ via the following proposition.

Proposition 3. *Take $\mathcal{T}$ as the η-dimensional standard grid triangulation on $V = \{0,1\}^\eta$, and a set $U \subseteq V$. For each $v \in U$, define $\Xi(v) \overset{\text{def}}{=} \{\, i \in [\![\eta]\!] \mid v_i = 0 \,\}$. The following are equivalent:*

1. *U is an infeasible set.*

2. *There does not exist an ordering $(v^{i_1}, \ldots, v^{i_{|U|}})$ of U such that*

$$\Xi(v^{i_1}) \subseteq \Xi(v^{i_2}) \subseteq \cdots \subseteq \Xi(v^{i_{|U|}}).$$

3. *There exists some $u, w \in U$ such that $\Xi(u) \nsubseteq \Xi(w)$ and $\Xi(w) \nsubseteq \Xi(u)$.*

Proof. $\underline{3 \implies 1}$ Condition 3 states that there are some $i, i' \in [\![\eta]\!]$ such that, w.l.o.g., $u^i = w^{i'} = 0$ and $u^{i'} = w^i = 1$. Therefore, $\{u, w\}$ is an infeasible set, since we cannot have that $x_i < x_{i'}$ (feasibility for u) and $x_{i'} < x_i$ (feasibility for w) simultaneously, as would be required for containment in one of the triangles T^π.

$\underline{1 \implies 2}$ Presume for contrapositive that such an ordering does exist, and assume w.l.o.g. that $i_j \equiv j$ for each $j \in [\![|U|]\!]$. Then we can select an permutation π of $[\![\eta]\!]$ such that

$$\begin{aligned} \{\pi(t)\}_{k=1}^{|\Xi(v^1)|} &= \Xi(v^1) \\ \{\pi(t)\}_{k=|\Xi(v^j)|+1}^{|\Xi(v^{j+1})|} &= \Xi(v^{j+1}) \backslash \Xi(v^j) \quad \forall j \in [\![|U|-1]\!] \\ \{\pi(t)\}_{k=|\Xi(v^{|U|})|+1}^{\eta} &= [\![\eta]\!] \backslash \left(\bigcup_{j=1}^{|U|} \Xi(v^j) \right). \end{aligned}$$

In words, this permutation orders the elements of $\Xi(v^1)$ first. The next set of elements in the permutation are those components $\Xi(v^2)\backslash\Xi(v^1)$ which are one for v^1, but zero for v^2. This is repeated for each j, leaving those not contained in any set until the end of the ordering. This verifies that $U \subseteq T^\pi$, verifying that U is a feasible set.

$\underline{2 \implies 3}$ Presume for contrapositive that no such $u, w \in U$ satisfying 3 exist. That is, for each $u, w \in U$, either $\Xi(u) \subseteq \Xi(w)$ or $\Xi(w) \subseteq \Xi(u)$. In other words, the sets $\{\Xi(u)\}_{u \in U}$ are nested, and we can produce an ordering $(v^{i_1}, \ldots, v^{i_{|U|}})$ satisfying condition 2. $\square$

Pairwise IB-representability follows from statement 3 in Proposition 3: each infeasible set $U \subseteq \{0,1\}^\eta$ can be reduced to an infeasible pair $\{u,w\} \subseteq U$.

2.4 Pairwise independent branching schemes

The pairwise independent branching scheme framework was initially introduced by Vielma and Nemhauser [135], where it was used to model particularly structured piecewise linear functions. In the remainder of this chapter, we will focus on pairwise IB schemes, and particularly at practical ways for constructing them.

2.4.1 Graphical representations of pairwise IB-representable CDCs

From our covering assumption $V = \bigcup\{T \in \mathcal{T}\}$, we can see that $|E| \geqslant 2$ for each $E \in \mathcal{E}_{\mathcal{T}}$. By applying Theorem 1, we then immediately have that $\mathrm{CDC}(\mathcal{T})$ is pairwise IB-representable if and only if $H^c_{\mathcal{T}}$ is (equivalent to) a graph. Along this line, for any constraint we may define a *conflict graph* for $\mathrm{CDC}(\mathcal{T})$ as $G^c_{\mathcal{T}} \overset{\mathsf{def}}{=} (V, \bar{E})$, where $\bar{E} \equiv \bar{E}_{\mathcal{T}} \overset{\mathsf{def}}{=} \{\, \{u,v\} \in [V]^2 \mid \{u,v\} \text{ is an infeasible set} \,\}$ is the set of all infeasible pairs of elements of V. Checking for pairwise IB-representability then reduces to verifying if $\mathcal{E}_{\mathcal{T}} = \bar{E}_{\mathcal{T}}$. The following corollary of Theorem 1 shows that this can also be verified by working only with $G^c_{\mathcal{T}}$.

Corollary 2. $\mathrm{CDC}(\mathcal{T})$ *is pairwise IB-representable if and only if the sets* $\mathcal{T}$ *are exactly the maximal independent sets of* $G^c_{\mathcal{T}}$.

Proof. If $\mathrm{CDC}(\mathcal{T})$ is pairwise IB-representable, then $G^c_{\mathcal{T}}$ is equivalent to $H^c_{\mathcal{T}}$. By applying Theorem 1, the maximal independent sets of $G^c_{\mathcal{T}}$ are exactly the elements of $\mathcal{T}$. For the converse, assume for a contradiction that $\mathcal{T}$ is exactly the maximal independent sets of $G^c_{\mathcal{T}}$, but that there exists some $E \in \mathcal{E}_{\mathcal{T}}$ with $|E| \geqslant 3$. By the minimal infeasibility of E, we have that $\{r,s\} \notin \bar{E}_{\mathcal{T}}$ for any distinct $r,s \in E$, and therefore E is an independent set in $G^c_{\mathcal{T}}$. This implies that E is contained in a maximal independent T. By assumption, $T \in \mathcal{T}$, which contradicts the infeasibility of E. $\square$

Therefore, verifying general pairwise IB-representability reduces to enumerating the maximal independent sets of $G^c_{\mathcal{T}}$ and identifying them to exactly the sets $\mathcal{T}$. As an example, we can see that, for the cardinality constraint of degree ℓ with $2 \leqslant \ell < N$ given by the family $\mathcal{T} \equiv \mathcal{T}^{\text{card}}_{N,\ell} = \{ T \subset [\![N]\!] \mid |T| = \ell \}$, the only maximal independent set of $G^c_{\mathcal{T}}$ is the entire ground set $V = [\![N]\!]$, which certainly cannot be identified with $\mathcal{T}$.

2.4.2 Representation at a given depth

Once a CDC has been shown to be pairwise IB-representable, a natural next question is: what is the smallest possible depth at which we may construct an IB scheme? Put another way , we ask if there exists a pairwise IB scheme for CDC($\mathcal{T}$) of some given depth t. The answer to this question reduces to the existence of a graphical decomposition of the conflict graph $G^c_{\mathcal{T}}$.

Definition 6. *A* biclique cover *of the graph $G = (V, E)$ is a collection of complete bipartite subgraphs $\{G^j = (V, E^j)\}_{j=1}^t$ of G that cover exactly the edges E of G. Formally, this means that there are some sets $\{(A^j, B^j)\}_{j=1}^t$ such that:*

- $\varnothing \subsetneq A^j, B^j \subsetneq V$ *for each* $j \in [\![t]\!]$,
- $A^j \cap B^j = \varnothing$ *for each* $j \in [\![t]\!]$,
- $E^j = A^j * B^j \overset{\text{def}}{=} \{ \{a, b\} \mid a \in A^j,\ b \in B^j \}$ *for each* $j \in [\![t]\!]$, *and*
- $\bigcup_{j=1}^t E^j = E$.

For notational simplicity, we will often refer to the sets $\{(A^j, B^j)\}_{j=1}^t$ as a biclique cover, as we can recover the graphs G^j directly.

The following theorem formalizes the equivalence between biclique covers and pairwise IB schemes.

Theorem 3. *If $\{(A^j, B^j)\}_{j=1}^t$ is biclique cover of the conflict graph $G^c_{\mathcal{T}}$ for pairwise IB-representable* CDC($\mathcal{T}$)*, then a pairwise IB scheme for* CDC($\mathcal{T}$) *is given by*

$$L^j = V \backslash A^j, \quad R^j = V \backslash B^j \qquad \forall j \in [\![t]\!]. \tag{2.8}$$

Conversely, if $\{(L^j, R^j)\}_{j=1}^t$ is a pairwise IB scheme for CDC($\mathcal{T}$)*, then a biclique cover of the conflict graph $G^c_{\mathcal{T}}$ is given by*

$$A^j = V \backslash L^j, \quad B^j = V \backslash R^j \qquad \forall j \in [\![t]\!]. \tag{2.9}$$

Proof. For the first part, take $\bar{E}$ as the edge set of $G^c_{\mathcal{T}}$. To see that any $T \in \mathcal{T}$ is feasible for the IB scheme (2.8), note that if $T \nsubseteq L^j$ and $T \nsubseteq R^j$, then there exist some $u \in A^j \cap T$ and $v \in B^j \cap T$. However, this implies that $\{u, v\} \in A^j * B^j \subseteq \bar{E}$, which is a contradiction of feasibility as $\{u, v\} \subseteq T$ and $T \in \mathcal{T}$. Furthermore, as $\{(A^j, B^j)\}_{j=1}^t$ is a biclique cover of $G^c_{\mathcal{T}}$, for every $\{u, v\} \in \bar{E}$ we have that there exists some level $j \in [\![t]\!]$ such that, w.l.o.g., $u \in A^j$ and $v \in B^j$. This implies that $u \notin L^j$ and $v \notin R^j$ by their construction, and as CDC($\mathcal{T}$) is pairwise IB-representable, then any infeasible set for CDC($\mathcal{T}$) is also infeasible for the proposed IB scheme. Therefore, (2.8) is a valid pairwise IB scheme.

For the second part, note that $A^j \cap B^j = \varnothing$ for all $j \in [\![t]\!]$, and that the covering portion of Assumption 2 implies that $L^j \cup R^j = V$. Therefore, it only remains to show that $\bar{E} = \bigcup_{j=1}^t \bar{E}^j$. For that, first note that as $L^j \cup R^j = V$, we have that $A^j = R^j \backslash L^j$ and $B^j = L^j \backslash R^j$. The containment $\bar{E} \subseteq \bigcup_{j=1}^t \bar{E}^j$ then follows by noting that, as $\{(L^j, R^j)\}_{j=1}^t$ is a valid pairwise IB scheme, each minimal infeasible set $\{u, v\} \in \bar{E}$ has some level $j \in [\![t]\!]$ such that $\{u, v\} \nsubseteq L^j$ and $\{u, v\} \nsubseteq R^j$. Then, as $L^j \cup R^j = V$, we have (w.l.o.g.) that $u \in L^j \backslash R^j \equiv B^j$ and $b \in R^j \backslash L^j \equiv A^j$, and so $\{a, b\} \in \bar{E}^j$. For the reverse containment $\bigcup_{j=1}^t \bar{E}^j \subseteq \bar{E}$, take some arbitrary $j \in [\![t]\!]$ and some edge $\{a, b\} \in \bar{E}^j$. From the definition of our biclique cover, we have that w.l.o.g. $a \in A^j \equiv R^j \backslash L^j$ and $b \in B^j \equiv L^j \backslash R^j$. Therefore, $\{a, b\}$ is an infeasible set for the IB scheme, and thus for CDC($\mathcal{T}$) as well, and so $\{a, b\} \in \bar{E}$. □

We can now naturally frame the problem of finding a minimum depth pairwise IB scheme as the minimum biclique cover problem [51, 57]. Unfortunately, the decision version of this problem is known to be NP-complete [109] and inapproximable within a factor of $|V|^{1/3-\epsilon}$ if $P \neq NP$ [61], even for bipartite graphs. However, we note that it is simple to construct a MIP feasibility problem for finding a pairwise IB scheme of a given depth t, which gives us a way to algorithmically find the smallest pairwise IB scheme for a specific (fixed) CDC.

Proposition 4. *A biclique cover of depth t exists for the conflict graph $G^c_{\mathcal{T}} = (V, \bar{E})$ of pairwise IB-representable* CDC($\mathcal{T}$) *if and only if the following admits a feasible solution:*

$$\left.\begin{aligned} z^{r,s}_j &\leqslant x^r_j + x^s_j \\ z^{r,s}_j &\leqslant x^r_j + y^r_j \\ z^{r,s}_j &\leqslant x^s_j + y^s_j \\ z^{r,s}_j &\leqslant y^r_j + y^s_j \\ z^{r,s}_j &\geqslant x^r_j + y^s_j - 1 \\ z^{r,s}_j &\geqslant x^s_j + y^r_j - 1 \end{aligned}\right\} \quad \forall j \in [\![t]\!], \forall \{r,s\} \in [V]^2 \tag{2.10a}$$

$$x^r_j + y^r_j \leqslant 1 \quad \forall j \in [\![t]\!], \forall r \in V \tag{2.10b}$$

$$\sum_{j=1}^{t} z^{r,s}_j \geqslant 1 \quad \forall \{r,s\} \in \bar{E} \tag{2.10c}$$

$$\sum_{j=1}^{t} z^{r,s}_j = 0 \quad \forall \{r,s\} \in [V]^2 \backslash \bar{E} \tag{2.10d}$$

$$x^r \in \{0,1\}^t \quad \forall r \in V \tag{2.10e}$$

$$y^r \in \{0,1\}^t \quad \forall r \in V \tag{2.10f}$$

$$z^{r,s} \in \{0,1\}^t \quad \forall \{r,s\} \in [V]^2. \tag{2.10g}$$

Moreover, for any feasible solution (x, y, z), a biclique cover for $G^c_{\mathcal{T}}$ is given by $A^j = \left\{ r \in V \;\middle|\; x^r_j = 1 \right\}$ and $B^j = \left\{ r \in V \;\middle|\; y^r_j = 1 \right\}$ for each $j \in [\![t]\!]$.

Proof. The interpretation of the decision variables is:

$$x_j^r = \mathbb{1}\left[r \in A^j\right] \tag{2.11a}$$

$$y_j^r = \mathbb{1}\left[r \in B^j\right] \tag{2.11b}$$

$$z_j^{r,s} = \mathbb{1}\left[x_j^r = y_j^s = 1 \text{ or } x_j^s = y_j^r = 1\right]. \tag{2.11c}$$

That is, $z_j^{r,s} = 1$ iff level i separates infeasible edge $\{r, s\} \in \bar{E}$, which is enforced via (2.10a-2.10b). To show that the existence of a biclique cover implies that (2.10) is feasible, you may consider the proposed solution (2.11) and see that it is feasible for (2.10).

To show that a feasible solution maps to a biclique cover, consider some (x, y, z) feasible for (2.10), and the corresponding sets $A^j = \left\{ r \in V \mid x_j^r = 1 \right\}$ and $B^j = \left\{ r \in V \mid y_j^r = 1 \right\}$ for each $j \in [\![t]\!]$. Inequalities (2.10b) ensure that $A^j \cap B^j = \varnothing$ for each $j \in [\![t]\!]$. Constraints (2.10d) ensure that $A^j * B^j \subseteq \bar{E}$ for each $j \in [\![t]\!]$. Therefore, each (A^j, B^j) is a biclique of $G^c_{\mathcal{T}}$. Furthermore, (2.10c) ensures that that there is at least one level j that separates each infeasible edge $\{r, s\} \in \bar{E}$. Therefore, $\{(A^j, B^j)\}_{j=1}^t$ is a biclique cover of $G^c_{\mathcal{T}}$. □

Additionally, Cornaz and Fonlupt [36] present a MIP formulation (with an exponential number of constraints that can be efficiently separated) to find the minimum level biclique cover of a graph.

We can now restate the MIP formulation from [135] (which is a special case of (2.3) with $k = 2$) in terms of biclique covers of $G^c_{\mathcal{T}}$.

Proposition 5 (Theorem 5, [135]; Theorem 1, [133])**.** *If* CDC($\mathcal{T}$) *is pairwise independent branching-representable and* $\{(A^j, B^j)\}_{j=1}^t$ *is a biclique cover for* $G^c_{\mathcal{T}}$*, then* *the following is an ideal formulation for* CDC($\mathcal{T}$)*:*

$$\sum_{v \in A^j} \lambda_v \leqslant z_j \qquad \forall j \in [\![t]\!] \tag{2.12a}$$

$$\sum_{v \in B^j} \lambda_v \leqslant 1 - z_j \qquad \forall j \in [\![t]\!] \tag{2.12b}$$

$$(\lambda, z) \in \Delta^V \times \{0,1\}^t. \tag{2.12c}$$

We end the section by noting that the relation between biclique covers and independent sets has also been exploited in the study of boolean functions, particularly in the equivalence between posiforms and maximum weighted stable sets (e.g. [37, Theorem 13.16]). In fact, formulation (2.12) is reminiscent of formulation (13.45–13.50) in [37, Theorem 13.13]. The main difference between these formulations is that in the context of [37] the λ variables will be binary variables not constrained to lie in the standard simplex. For this reason inequalities (2.12a–2.12b) appear disaggregated in [37, Theorem 13.13] in the form $\lambda_v \leqslant z_j$ for all $v \in A^j$, $j \in [\![t]\!]$. However, the resulting formulation is not ideal (See [135, Section 5] for more details). Still, the combinatorial aspects of this connection could prove useful for constructing small IB schemes.

For the remainder, we will explore instances where we can, in closed form, construct small (asymptotically optimal) IB schemes for families of particularly structured CDCs. We will apply our methodology to these specific structures, and produce small, closed-form IB schemes. In particular, this allows us to construct novel, small MIP formulations for these constraints.

2.5 A simple IB scheme and its limitations

To start, we show that *any* pairwise IB-representable CDC admits an IB scheme of depth $|V|$. If $|V|$ is smaller than $|\mathcal{T}|$, this already offers a drop in size from (1.17). This IB scheme covers all edges incident to node with the simple biclique corresponding to the star centered at that node.

Proposition 6 (Covering with Stars)**.** *For pairwise IB-representable* CDC($\mathcal{T}$)*, a* *biclique cover for $G^c_{\mathcal{T}} = (V, \bar{E})$ is given by:*

$$A^v = \{v\}, \qquad B^v = \left\{ u \in V \;\middle|\; \{u, v\} \in \bar{E} \right\} \qquad \forall v \in V.$$

Proof. By construction of the sets, we see that each $\{r, s\} \in \bar{E}^v \equiv A^v * B^v$ corresponds to an infeasible edge: that is, $\bar{E}^v \subseteq \bar{E}$ for each v, and so $\bigcup_{v\in V} \bar{E}^v \subseteq \bar{E}$. Furthermore, each infeasible edge $\{r, s\} \in \bar{E}$ is infeasible for levels r and s, and so $\bar{E} \subseteq \bigcup_{v\in V} \bar{E}^v$. Therefore, this construction forms a valid biclique cover of the conflict graph. $\square$

This gives us an upper bound of $|V|$ on the minimum depth for any pairwise IB-representable CDC. However, if we exploit the specific structure of a CDC, we can typically get much smaller formulations. For instance, consider the following two instances of the SOS3(N) constraint for small values of N. First, consider the instance with $N = 6$, where $|V| = 6$ and $\mathcal{T} = (\{1, 2, 3\}, \{2, 3, 4\}, \{3, 4, 5\}, \{4, 5, 6\})$. Therefore, $|\mathcal{T}| = 4$, yielding a lower bound of depth $\log_2(4) = 2$ from Proposition 1. However, there does not exist a biclique cover of depth 2 (which can be verified via Proposition 4), though one of depth 3 does exist:

$$\begin{aligned} A^1 &= \{1\}, & B^1 &= \{4, 5, 6\} \\ A^2 &= \{1, 2\}, & B^2 &= \{5, 6\} \\ A^3 &= \{1, 2, 3\}, & B^3 &= \{6\}. \end{aligned}$$

We can see the proposed IB scheme on the left side of Figure 2-3. For clarity, the associated MIP formulation for the CDC from Proposition 5 is

$$\begin{aligned} \lambda_1 &\leqslant z_1 & \lambda_4 + \lambda_5 + \lambda_6 &\leqslant 1 - z_1 \\ \lambda_1 + \lambda_2 &\leqslant z_2 & \lambda_5 + \lambda_6 &\leqslant 1 - z_2 \\ \lambda_1 + \lambda_2 + \lambda_3 &\leqslant z_3 & \lambda_6 &\leqslant 1 - z_3 \\ (\lambda, z) &\in \Delta^6 \times \{0, 1\}^3. \end{aligned}$$

Next, we consider $N = 10$, where we also cannot attain the $\log_2(8) = 3$ lower bound. However, a biclique for this the conflict graph of this constraint is

$$
\begin{aligned}
A^1 &= \{1,8,9,10\}, & B^1 &= \{4,5\} && \text{(2.14a)}\\
A^2 &= \{1,2,10\}, & B^2 &= \{5,6,7\} && \text{(2.14b)}\\
A^3 &= \{1,2,3,9,10\}, & B^3 &= \{6\} && \text{(2.14c)}\\
A^4 &= \{1,2,3,4\}, & B^4 &= \{7,8,9,10\}, && \text{(2.14d)}
\end{aligned}
$$

as seen on the right side of Figure 2-3. The corresponding MIP formulation is

$$
\begin{aligned}
\lambda_1 + \lambda_8 + \lambda_9 + \lambda_{10} &\leqslant z_1 & \lambda_4 + \lambda_5 &\leqslant 1 - z_1\\
\lambda_1 + \lambda_2 + \lambda_{10} &\leqslant z_2 & \lambda_5 + \lambda_6 + \lambda_7 &\leqslant 1 - z_2\\
\lambda_1 + \lambda_2 + \lambda_3 + \lambda_9 + \lambda_{10} &\leqslant z_3 & \lambda_6 &\leqslant 1 - z_3\\
\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 &\leqslant z_4 & \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} &\leqslant 1 - z_4\\
& (\lambda, z) \in \Delta^{10} \times \{0,1\}^4.
\end{aligned}
$$

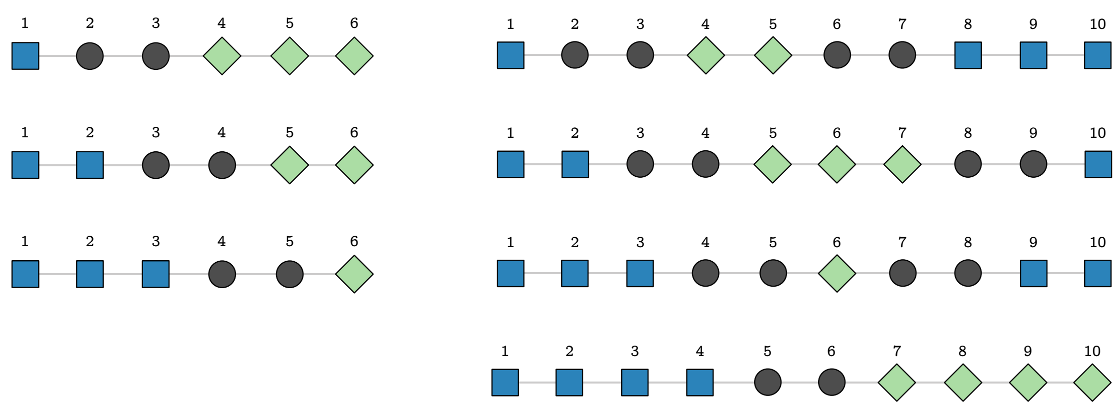


Figure 2-3: Visualizations of the biclique covers presented in the text for **(Left)** SOS3(6) and **(Right)** SOS3(10). Each row corresponds to some level j, and the elements of A^j and B^j are the blue squares and green diamonds, respectively.

The ad-hoc construction for SOS3(6) suggests a more general construction for SOS$k(N)$ when $k \leqslant N/2$ (assume for convenience that N is even). Consider the sets given by

$$A^j = \{1, \ldots, j\} \cup \{j + N/2 + k, \ldots, N\}, \qquad B^j = \{j + k, \ldots, j + N/2\}$$

for each $j \in [\![N/2]\!]$. It is straightforward to see that this yields a biclique cover of the conflict graph for SOS$k(N)$ of depth $N/2$. Therefore, with this simple operation, we have constructed an ideal formulation for SOS$k(N)$ with size strictly smaller than N, the size of the non-extended formulation (1.17).

Based on the second example (2.14), we know that this construction is, in general, not the smallest possible. In Chapter 2.8.4, we will see how we can systematically construct small biclique covers (and MIP formulations) for SOS$k(N)$ with arbitrary k and N, using techniques we will now develop.

2.6 Systematic construction of biclique covers

As discussed in Chapter 2.3.4, there exists an IB scheme for the SOS2 constraint of optimal depth that can be constructed using a Gray code. The following proposition shows how the validity of this scheme can easily be proven by reinterpreting it through a biclique cover.

Proposition 7. *Take $N = d+1$, a Gray code $\{h^i\}_{i=1}^d \subseteq \{0,1\}^{\lceil \log_2(d) \rceil}$, and let $h^0 \stackrel{\text{def}}{=} h^1$ and $h^{d+1} \stackrel{\text{def}}{=} h^d$. If $\mathcal{T} \equiv \mathcal{T}_d^{\text{SOS2}}$ and $G^c_{\mathcal{T}}$ is the conflict graph of SOS2(N), then a biclique cover for $G^c_{\mathcal{T}}$ of depth $\lceil \log_2(d) \rceil$ is given by*

$$A^j = \left\{ \tau \in [\![N]\!] \;\middle|\; h_j^{\tau-1} = h_j^\tau = 0 \right\} \tag{2.15a}$$

$$B^j = \left\{ \tau \in [\![N]\!] \;\middle|\; h_j^{\tau-1} = h_j^\tau = 1 \right\} \tag{2.15b}$$

for all $j \in [\![\lceil \log_2(d) \rceil]\!]$.

Proof. For the SOS2(N) constraint we have that $\bar{E}_{\mathcal{T}} = \{ \{r, s\} \in [\![N]\!]^2 \mid r + 2 \leqslant s \}$. Take any infeasible pair $\{r, s\} \in \bar{E}_{\mathcal{T}}$. As $r + 2 \leqslant s$, we conclude that $r - 1 < r < s - 1 < s$, and so it must be that $h^{r-1}, h^r \neq h^{s-1}, h^s$. The set of components which flip values between the two pairs of adjacent codes (h^{r-1}, h^r) and (h^{s-1}, h^s) is $I = \left\{ j \in [\![\lceil \log_2(d) \rceil]\!] \;\middle|\; h^{r-1}_j \neq h^r_j \text{ or } h^{s-1}_j \neq h^s_j \right\}$, and $|I| \leqslant 2$ as we have selected a Gray code. Now it must be the case that there is some component $j \in [\![\lceil \log_2(d) \rceil]\!] \backslash I$ wherein $h^{r-1}_j = h^r_j \neq h^{s-1}_j = h^s_j$, else we conclude that two of the vectors $h^i = h^\ell$ coincide for some $i \in \{r-1, r\}$ and $\ell \in \{s-1, s\}$, a contradiction of their uniqueness. Then $\{r, s\} \in E^j$, i.e. it is covered by the j-th level of the biclique. Furthermore, we observe that no edges of the form $\{r, r+1\}$ will be contained in the biclique cover, as it is not possible that $h^{r-1}_j = h^r_j = 0$ (resp. $= 1$) and $h^r_j = h^{r+1}_j = 1$ (resp. $= 0$) simultaneously. $\square$

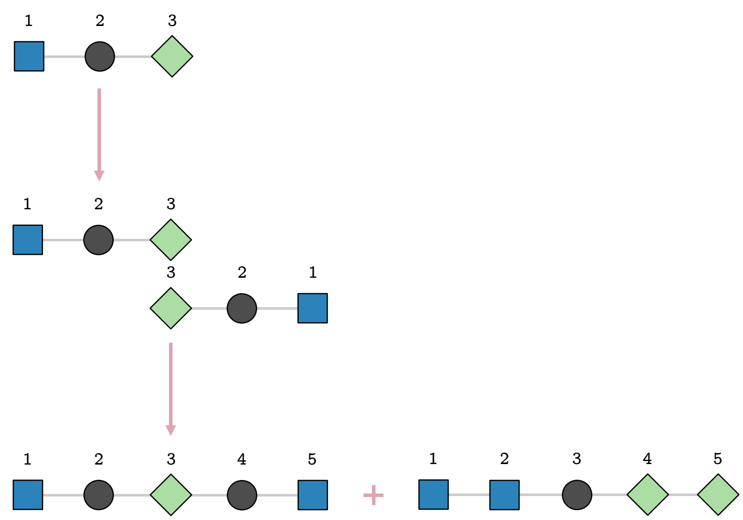


Figure 2-4: The recursive construction for biclique covers for SOS2. The first row is a single biclique that covers the conflict graph for SOS2(3) ($A^{1,1}$ in blue, $B^{1,1}$ in green). The second row shows the construction which duplicates the ground set $\{1, 2, 3\}$ and inverts the ordering on the second copy. The third row shows the identification of the nodes that yields a valid biclique for SOS2(5). This biclique is then combined with a second that covers all edges between nodes identified with the first copy and those identified with the second, giving a biclique cover for SOS2(5) with two levels.

Interestingly, we can also view this construction recursively if we use a specific Gray code known as the binary reflected Gray code [120]. For SOS2(2^k), we will take $\bar{E}^k$ as the edge set for the corresponding conflict graph. First, with $k = 1$, $d = 2^k = 2$, and $N = 2^k + 1 = 3$, then $\bar{E}^1 = \{\{1, 3\}\}$. A complete biclique cover is given by the single biclique $A^{1,1} = \{1\}$ and $B^{1,1} = \{3\}$. As we see in Figure 2-4, we can construct a biclique cover for SOS2(5) (i.e. $k = 2$) by stitching together two copies of the biclique $(A^{1,1}, B^{1,1})$ in the following way. We construct two copies of the node set for $k = 1$, invert the second, and identify the last node from the first set with the first node with the second set. Then we can readily construct a mapping of the biclique $(A^{1,1}, B^{1,1})$ for $k = 1$ to a biclique for $k = 2$, using the node identification, as $A^{2,1} = \{1, 5\}$ and $B^{2,1} = \{3\}$. This will cover all edges in $\bar{E}^2$ with both incident nodes in the first half of the nodes, or both in the second half of the nodes (along with some other edges in $\bar{E}^2$, as well). To cover all edges with one adjacent node in the first half, and the other in the second half, we construct a second biclique of the form $A^{2,2} = \{1, 2\}$ and $B^{2,2} = \{4, 5\}$.

We can repeat this construction with $k = 3$ to get the three level biclique cover

$$\begin{aligned}
A^{3,1} &= \{1, 5, 9\}, & B^{3,1} &= \{3, 7\} \\
A^{3,2} &= \{1, 2, 8, 9\}, & B^{3,2} &= \{4, 5, 6\} \\
A^{3,3} &= \{1, 2, 3, 4\}, & B^{3,2} &= \{6, 7, 8, 9\}.
\end{aligned}$$

Applying this repeatedly yields a biclique cover for $\bar{E}^{k+1}$ as $\{(A^{k+1,i}, B^{k+1,i})\}_{i=1}^{k+1}$, where

$$\begin{aligned}
A^{k+1,i} &= \bigcup_{u \in A^{k,i}} \left\{u, 2^{k+1} + 2 - u\right\}, & B^{k+1,i} &= \bigcup_{v \in B^{k,i}} \left\{v, 2^{k+1} + 2 - v\right\} \quad \forall i \in [\![k]\!] \\
A^{k+1,k+1} &= \left\{1, \ldots, 2^k\right\}, & B^{k+1,k+1} &= \left\{2^k + 2, \ldots, 2^{k+1} + 1\right\}.
\end{aligned}$$

We will refer to the resulting formulation via (5) as the *logarithmic independent branching* (`LogIB`) formulation for the SOS2 constraint, and observe that it coincides with the logarithmic formulation presented by Vielma and Nemhauser [133]. Additionally, we note that it readily generalizes to the case where d is not a power-of-two.

Additionally, we can readily state this recursive construction in a more general form, where we adapt a biclique cover for one graph into a biclique cover for another graph that is created in some specific way.

Lemma 2. *Take some graph $G = (\llbracket m+1 \rrbracket, E)$, and define $G^2 = (\llbracket 2m+1 \rrbracket, E^2)$, where*

$$E^2 = E \cup \{\, \{2m+2-u, 2m+2-v\} \mid \{u,v\} \in E \,\} \cup \left(\llbracket m \rrbracket * \llbracket m+2, 2m+1 \rrbracket \right)$$

where $\llbracket a,b \rrbracket \stackrel{\mathsf{def}}{=} \{a,\dots,b\}$. If $\{(A^j,B^j)\}_{j=1}^t$ is a biclique cover of G, then $\{(\tilde{A}^j,\tilde{B}^j)\}_{j=1}^{t+1}$ is a biclique cover of G^2, where

$$\begin{aligned} \tilde{A}^j &= \bigcup_{u \in A^j} \{u, 2m+2-u\}, & \tilde{B}^j &= \bigcup_{v \in B^j} \{v, 2m+2-v\} \quad \forall j \in \llbracket t \rrbracket \\ \tilde{A}^{t+1} &= \{1,\dots,m\}, & \tilde{B}^{t+1} &= \{m+2,\dots,2m+1\}. \end{aligned}$$

In the remainder of this work, we will see how we may apply similar graphical results to systematically construct small biclique covers for the conflict graphs of constraints by exploiting their specific structure.

2.7 Biclique covers for graph products and discretizations of multilinear functions

Consider the discretization of multilinear functions described in Chapter 1.3.5, given by $V = \prod_{i=1}^{\eta} \llbracket N_i \rrbracket$ and $\mathcal{T} = (\prod_{i=1}^{\eta} \{k_i, k_i+1\} \mid k \in \prod_{i=1}^{\eta} \llbracket d_i \rrbracket)$ (recall that $N_i = d_i + 1$ for each i). We can interpret this constraint as a η-dimensional version of the SOS2 constraint, or as the Cartesian product of η SOS2 constraints. This can be formalized through the following definition and straightforward lemma.

Definition 7. *The* (disjunctive) graph product *of a family of graphs $\{G^i = (V^i, E^i)\}_{i=1}^{\eta}$ is $\bigvee_{i=1}^{\eta} G^i \stackrel{\mathsf{def}}{=} (V_P, E_P)$, where $V_P = \prod_{i=1}^{\eta} V^i$ and*

$$E_P = \left\{ \{u,v\} \in [V_P]^2 \;\middle|\; \exists i \in \llbracket \eta \rrbracket \text{ s.t. } \{u_i, v_i\} \in E^i \right\}.$$

Lemma 3. *Let $V = \prod_{i=1}^{\eta} \llbracket N_i \rrbracket$ and $\mathcal{T} = (\prod_{i=1}^{\eta} \{k_i, k_i+1\} \mid k \in \prod_{i=1}^{\eta} \llbracket d_i \rrbracket)$ be a η-dimensional discretization of a multilinear function, and $G^c_{\mathcal{T}}$ be the corresponding* *conflict graph. If G^i is the conflict graph of SOS2(N_i) for each $i \in [\![\eta]\!]$, then $G^c_{\mathcal{T}} = \bigvee_{i=1}^{\eta} G^i$.*

Using this characterization, we can easily construct an IB scheme for discretizations of multilinear functions by taking the graph products of IB schemes for the SOS2 constraint.

Lemma 4. *Take a family of graphs $\{G^i = (V^i, E^i)\}_{i=1}^{\eta}$, and a biclique cover $\{(\tilde{A}^{i,j}, \tilde{B}^{i,j})\}_{j=1}^{t_i}$ for each G^i. Then a biclique cover for $\bigvee_{i=1}^{\eta} G^i$ is given by $\bigcup_{i=1}^{\eta}\{(A^{i,j}, B^{i,j})\}_{j=1}^{t_i}$, where*

$$A^{i,j} = \left(\prod_{\ell=1}^{i-1} V^{\ell}\right) \times \tilde{A}^{i,j} \times \left(\prod_{\ell=i+1}^{\eta} V^{\ell}\right)$$
$$B^{i,j} = \left(\prod_{\ell=1}^{i-1} V^{\ell}\right) \times \tilde{B}^{i,j} \times \left(\prod_{\ell=i+1}^{\eta} V^{\ell}\right)$$

for all $i \in [\![\eta]\!]$ and $j \in [\![t_i]\!]$.

Corollary 3. *Let $V = \prod_{i=1}^{\eta}[\![N_i]\!]$ and $\mathcal{T} = (\prod_{i=1}^{\eta}\{k_i, k_i + 1\} \mid k \in \prod_{i=1}^{\eta}[\![d_i]\!])$ describe a η-dimensional discretization of a multilinear function, and take $G^c_{\mathcal{T}}$ as its conflict graph. If for each $i \in [\![\eta]\!]$ we have a biclique cover $\{(\tilde{A}^{i,j}, \tilde{B}^{i,j})\}_{j=1}^{t_i}$ for the conflict graph of SOS2(N_i), then a biclique cover for $G^c_{\mathcal{T}}$ of depth $\sum_{i=1}^{\eta} t_i$ is given by $\bigcup_{i=1}^{\eta}\{(A^{i,j}, B^{i,j})\}_{j=1}^{t_i}$, where*

$$A^{i,j} = \left\{ x \in V \;\middle|\; x_i \in \tilde{A}^{i,j} \right\}, \quad B^{i,j} = \left\{ x \in V \;\middle|\; x_i \in \tilde{B}^{i,j} \right\} \qquad \forall i \in [\![\eta]\!], j \in [\![t_i]\!].$$

In particular, if we take $\{h^{i,j}\}_{j=1}^{d_i} \subseteq \{0,1\}^{\lceil \log_2(d_i) \rceil}$ as a Gray code for each $i \in [\![\eta]\!]$, where $h^{i,0} \equiv h^{i,1}$ and $h^{i,d_i+1} \equiv h^{i,d_i}$, then a biclique cover for $G^c_{\mathcal{T}}$ of depth $\sum_{i=1}^{\eta}\lceil \log_2 d_i \rceil$ is given by:

$$A^{i,j} = \left\{ x \in V \mid \exists \gamma \text{ s.t. } x_i = \gamma,\ h_j^{i,\gamma-1} = h_j^{i,\gamma} = 0 \right\}$$
$$B^{i,j} = \left\{ x \in V \mid \exists \gamma \text{ s.t. } x_i = \gamma,\ h_j^{i,\gamma-1} = h_j^{i,\gamma} = 1 \right\}$$

for each $i \in [\![\eta]\!]$ and $j \in [\![\lceil \log_2(d_i) \rceil]\!]$.

We note that, since $|\mathcal{T}| = \prod_{i=1}^{\eta} d_i$, by Proposition 1 this construction yields a formulation that is asymptotically optimal (with respect to number of binary variables) for any possible binary MIP formulation, up to an additive factor of at most η.

Furthermore, we can specialize this to the bilinear case studied by Misener et al. [106].

Corollary 4. *There exists a biclique cover for a the grid discretization of a bilinear function ($\eta = 2$) with $d_2 = 0$ of depth $\lceil \log_2(d_1) \rceil$.*

This result yields an ideal MIP formulation for the outer-approximation of bilinear terms with $\lceil \log_2(d_1) \rceil$ binary variables, $2(d_1 + 1)$ auxiliary continuous variables (the λ variables, one for element in V), and $2\lceil \log_2(d_1) \rceil$ general inequality constraints. In contrast, the logarithmic formulation from Misener et al. [106] has $\lceil \log_2(d_1) \rceil$ binary variables, $2\lceil \log_2(d_1) \rceil + 1$ auxiliary continuous variables, at least $2\lceil \log_2(d_1) \rceil + 6$ general inequality constraints, and is not ideal in general (see Appendix A). Therefore, we gain an ideal formulation with a naturally induced constraint branching at the price of a modest number of additional auxiliary continuous variables. Furthermore, our formulation generalizes readily to discretization along the second dimension ($d_2 \geqslant 1$), for non-uniform discretizations, and for higher dimensional multilinear functions ($\eta > 2$).

2.8 Completing biclique covers via graph unions

Another useful graphical technique for our heuristic constructions will be to combine together biclique covers, each of which is designed to cover a substructure of the constraint. For example, the conflict graph of a grid triangulation of the plane is equivalent to the conflict graph of a 2-dimensional grid discretization of a multilinear function, with one extra edge added for each subrectangle in the grid. Therefore, a biclique cover of a grid triangulation can be obtained from a biclique cover of a 2-dimensional discretization of a multilinear function (i.e. from Corollary 3) by completing it with some number additional bicliques that cover those extra edges.

This construction can be formalized in the following way.

Definition 8. *The* graph union *of a family of graphs* $\{G^i = (V^i, E^i)\}_{i=1}^{\eta}$ *is* $\bigcup_{i=1}^{\eta} G^i \stackrel{\text{def}}{=} (V_U, E_U)$*, where* $V_U = \bigcup_{i=1}^{\eta} V^i$ *and* $E_U = \bigcup_{i=1}^{\eta} E^i$*.*

Lemma 5. *Take a family of graphs* $\{G^i = (V^i, E^i)\}_{i=1}^{\eta}$ *and a corresponding biclique cover* $\{(A^{i,j}, B^{i,j})\}_{j=1}^{t_i}$ *of* G^i *for each* $i \in [\![\eta]\!]$*. Then* $\bigcup_{i=1}^{\eta}\{(A^{i,j}, B^{i,j})\}_{j=1}^{t_i}$ *is a biclique cover of* $\bigcup_{i=1}^{\eta} G^i$*.*

We can apply Lemma 5 to construct biclique covers for the grid triangulations depicted in Figure 1-2. First, we apply the biclique cover construction from Corollary 3 to cover all edges not sharing a subrectangle. This is depicted in the first two subfigures of each row in Figure 2-5. To cover the remaining 4 edges created by the triangulation, we see that the number of additional levels needed is dependent on the combinatorial structure. Additionally, in all three cases we can verify through Proposition 4 that the resulting biclique cover is of the smallest possible depth. The first

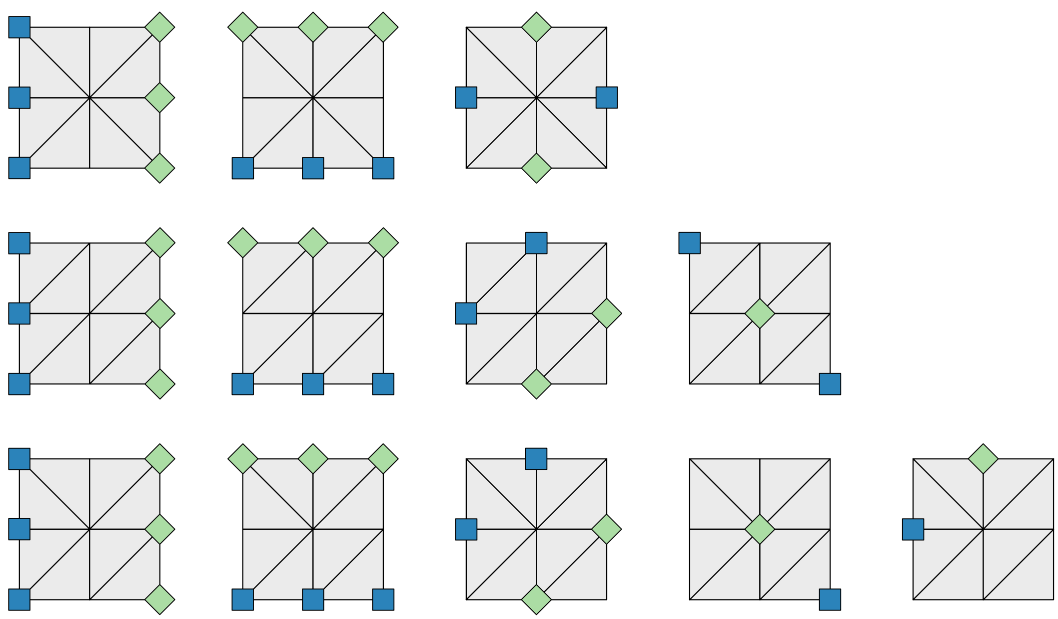


Figure 2-5: Independent branching schemes for the three triangulations presented in Figure 1-2, each given its own row.The sets A^j and B^j are given by the blue squares and green diamonds, respectively, in the i-th subfigure of the corresponding row.

example is the "Union Jack" triangulation [127] for $d_1 = d_2 = 2$, where the results of Vielma and Nemhauser [135] show that for this triangulation the biclique cover from Corollary 3 can be completed with a single additional biclique cover for any d_1 and d_2. The second triangulation is a K1 triangulation [84] for $d_1 = d_2 = 2$, and an early version of [131] showed that for this triangulation the biclique can always be completed with two additional bicliques.

In contrast, for generic triangulations such as the third one, it was not previously known if the biclique cover can always be completed with fewer than the trivial $d_1 \cdot d_2$ levels needed to cover each "diagonal" edge with its own additional biclique. First, we can adapt Proposition 6 to cover the extra edges with stars, but in general this will result in $\Theta(d_1 \cdot d_2)$ stars, and hence the same number of additional levels. To reduce this, we need a way to *stick* the stars together into more complicated bicliques. It is also possible to use graph colorings for a certain class of triangulations (subsuming the Union Jack and K1 triangulations as special cases), to cover the extra edges with either one or two additional bicliques [67]. In general, it turns out that we may cover the remaining edges for any grid triangulation with a constant number of additional levels by applying the following simple lemma.

Lemma 6. *Let $\{(A^j, B^j)\}_{j=1}^t$ be a family of bicliques of a graph G. If (A^k, B^ℓ) is also a biclique of G for each $k, \ell \in [\![t]\!]$, then $\left(\bigcup_{j=1}^t A^j, \bigcup_{j=1}^t B^j\right)$ is a biclique of G.*

The strength of Lemma 6 comes from the fact that many CDCs of practical interest have a local structure (i.e. sets in $\mathcal{T}$ have small cardinality, or, equivalently, the minimum degree of the conflict graph is close to the total number of nodes). In this case, the condition of Lemma 6 will hold for families of stars centered at nodes that are located "sufficiently far apart."

2.8.1 Grid triangulations of the plane

We may now present a biclique cover construction for generic grid triangulations, with no further assumptions on the structure of the triangles such as in [133, 135], whose depth scales like $\log_2(d_1) + \log_2(d_2) + \mathcal{O}(1)$. In the same way as depicted in Figure 2-5, we construct the biclique cover by using Lemma 5 to complete the construction of

Corollary 3. For this, we will use the following corollary of Lemma 6 that shows how to combine certain stars centered at sufficiently separated nodes.

Corollary 5. *Take a regular grid $V = [\![N_1]\!] \times [\![N_2]\!]$ where $N_1 \equiv d_1 + 1$ and $N_2 \equiv d_2 + 1$, let $\mathcal{T}$ be a grid triangulation of $[1, N_1] \times [1, N_2]$, and take $G^c_{\mathcal{T}} = (V, \bar{E})$ as its conflict graph. Define $A(w) \overset{\text{def}}{=} \{w\}$ and $B(w) \overset{\text{def}}{=} \left\{ w + v \mid v \in \{-1, 1\}^2, \{w, w + v\} \in \bar{E} \right\}$ for each $w \in V$. Then*

$$\left(\bigcup_{w \in V \cap (u + 3\mathbb{Z}^2)} A(w), \bigcup_{w \in V \cap (u + 3\mathbb{Z}^2)} B(w) \right)$$

is a biclique of $G^c_{\mathcal{T}}$ for any $u \in V$.

Proof. Direct from Lemma 6 by taking $u \in V$ and the family of bicliques

$$\{(A(w), B(w)\}_{w \in V \cap (u + 3\mathbb{Z}^2)}$$

and noting that, if $u, v \in V \cap (u + 3\mathbb{Z}^2)$, then $||u - v||_\infty \geqslant 3$, and so $(A(u), B(v))$ is also a biclique for $G^c_{\mathcal{T}}$. □

Figure 2-6 shows two possible bicliques that can be obtained from Corollary 5.

We can now use Lemma 5 and Corollary 5, along with the biclique cover derived in Corollary 3, to obtain a biclique cover for any triangulation with an asymptotically optimal number of levels.

Theorem 4. *Take $V = [\![N_1]\!] \times [\![N_2]\!]$ where $N_1 \equiv d_1 + 1$ and $N_2 \equiv d_2 + 1$, and let $\mathcal{T}$ be a grid triangulation of $[1, N_1] \times [1, N_2]$. Take $G^c_{\mathcal{T}} = (V, \bar{E})$ as its conflict graph. Presume that $\{(\tilde{A}^{1,j}, \tilde{B}^{1,j})\}_{j=1}^{t_1}$ and $\{(\tilde{A}^{2,j}, \tilde{B}^{2,j})\}_{j=1}^{t_2}$ are biclique covers for the conflict graphs of the SOS2(N_1) and SOS2(N_2) constraints, respectively. Furthermore, define*

$$\begin{aligned} A^{3,u} &= V \cap (u + 3\mathbb{Z}^2) \\ B^{3,u} &= \bigcup_{w \in V \cap (u + 3\mathbb{Z}^2)} \left\{ w + v \mid v \in \{-1, 1\}^2, \ \{w, w + v\} \in \bar{E} \right\} \end{aligned}$$

for each $u \in \{0, 1, 2\}^2$. Then $\{(A^{1,j}, B^{1,j})\}_{j=1}^{t_1} \cup \{(A^{2,j}, B^{2,j})\}_{j=1}^{t_2} \cup \{(A^{3,u}, B^{3,u})\}_{u \in \{0,1,2\}^\eta}$

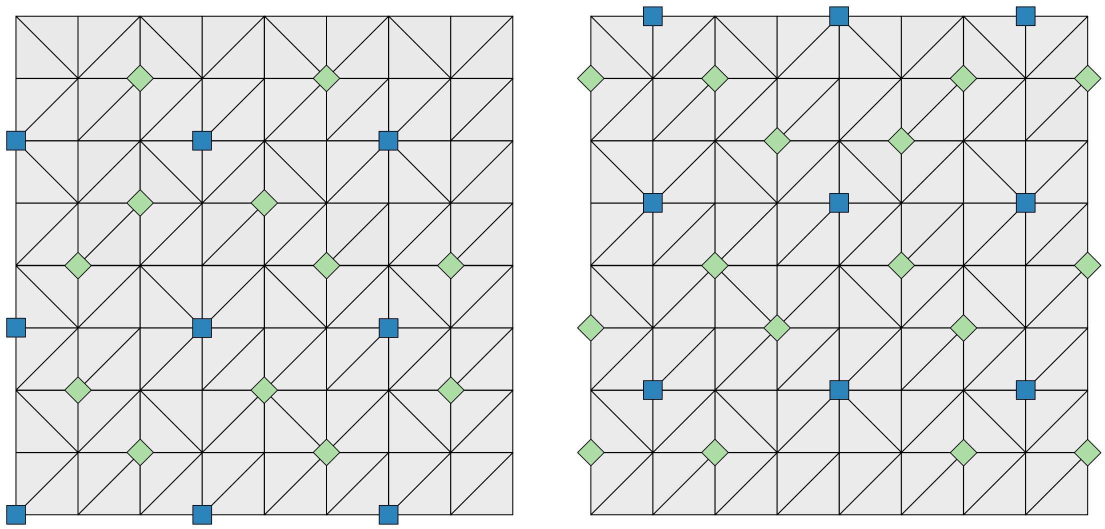


Figure 2-6: Two bicliques constructed via Corollary 5 for a grid triangulation with $d_1 = d_2 = 8$. On the left the construction follows by taking $u = (1, 1)$; on the right, with $u = (2, 3)$. For each level, the sets $A(u)$ and $B(u)$ are given by the blue squares and green diamonds, respectively.

is a biclique cover for $G^c_{\mathcal{T}}$, where

$$\begin{aligned} A^{1,j} &= \tilde{A}^{1,j} \times [\![N_2]\!], & B^{1,j} &= \tilde{B}^{1,j} \times [\![N_2]\!], \\ A^{2,j'} &= [\![N_1]\!] \times \tilde{A}^{2,j'}, & B^{2,j'} &= [\![N_1]\!] \times \tilde{B}^{2,j'}, \end{aligned}$$

for each $j \in [\![t_1]\!]$ and $j' \in [\![t_2]\!]$.

In particular, if $\{h^{1,i}\}_{i=1}^{d_1} \subseteq \{0,1\}^{\lceil \log_2(d_1) \rceil}$ and $\{h^{2,i}\}_{i=1}^{d_2} \subseteq \{0,1\}^{\lceil \log_2(d_2) \rceil}$ are Gray codes, where $h^{1,0} \stackrel{\text{def}}{=} h^{1,1}$, $h^{1,d_1+1} \stackrel{\text{def}}{=} h^{1,d_1}$, $h^{2,0} \stackrel{\text{def}}{=} h^{2,1}$, and $h^{2,d_2+1} \stackrel{\text{def}}{=} h^{2,d_2}$, then a *biclique cover of $G^c_{\mathcal{T}}$ of depth $\lceil \log_2(d_1) \rceil + \lceil \log_2(d_2) \rceil + 9$ is given by:*

$$A^{1,j} = \left\{ (x,y) \in V \;\middle|\; h^{1,x-1}_j = h^{1,x}_j = 1 \right\} \tag{2.16a}$$
$$B^{1,j} = \left\{ (x,y) \in V \;\middle|\; h^{1,x-1}_j = h^{1,x}_j = 0 \right\} \tag{2.16b}$$
$$A^{2,j'} = \left\{ (x,y) \in V \;\middle|\; h^{2,y-1}_{j'} = h^{2,y}_{j'} = 1 \right\} \tag{2.16c}$$
$$B^{2,j'} = \left\{ (x,y) \in V \;\middle|\; h^{2,y-1}_{j'} = h^{2,y}_{j'} = 0 \right\} \tag{2.16d}$$
$$A^{3,u} = V \cap (u + 3\mathbb{Z}^2) \tag{2.16e}$$
$$B^{3,u} = \bigcup_{w \in V \cap (u+3\mathbb{Z}^2)} \left\{ w + v \;\middle|\; v \in \{-1,1\}^2,\ \{w, w+v\} \in \bar{E} \right\} \tag{2.16f}$$

for all $j \in [\![\lceil \log_2(d_1) \rceil]\!]$, $j' \in [\![\lceil \log_2(d_2) \rceil]\!]$, and $u \in \{0,1,2\}^2$.

Proof. Let $G^x \stackrel{\text{def}}{=} ([\![N_1]\!], E^x)$ and $G^y \stackrel{\text{def}}{=} ([\![N_2]\!], E^y)$ be the conflict graphs for SOS2(N_1) and SOS2(N_2), respectively. Furthermore, let

$$G^3 \stackrel{\text{def}}{=} \bigcup_{u \in \{0,1,2\}^2} (V, A^{3,u} * B^{3,u}) = \left(V, \bigcup_{u \in \{0,1,2\}^2} (A^{3,u} * B^{3,u}) \right).$$

Then we see that $G_{\mathcal{T}} = (G^x \times G^y) \cup G^3$ by noting that all *diagonal* edges of E (i.e. those of the form $\{w, w+v\} \in E$ for $w \in V$ and $v \in \{-1,1\}^2$) are included in G^3, and observing that G^3 is a subgraph of $G^c_{\mathcal{T}}$. The result then follows from Lemma 4, Lemma 5, and Corollary 5. □

By referring to Proposition 1, we recover a $\lceil \log_2(2d_1 \cdot d_2) \rceil \geqslant \lceil \log_2(d_1) \rceil + \lceil \log_2(d_2) \rceil$ lower bound on the depth of any biclique cover for a grid triangulation, and see that our construction yields a MIP formulation that is within a constant additive factor of the smallest possible.

2.8.2 Higher-dimensional grid triangulations

We can see that the "stencil" construction for bivariate grid triangulations readily generalizes to higher-dimensional grid triangulations as well.

Theorem 5. *Take $V = \prod_{i=1}^{\eta} [\![N_i]\!]$ where $N_1 \equiv d_1 + 1$ and $N_2 \equiv d_2 + 1$, and let $\mathcal{T}$ be a grid triangulation of $\prod_{i=1}^{\eta}[1, N_i]$. Take $G^c_{\mathcal{T}} = (V, \bar{E})$ as its conflict graph. For each $i \in [\![\eta]\!]$, presume that $\{(\tilde{A}^{i,j}, \tilde{B}^{i,j})\}_{j=1}^{t_1}$ is a biclique cover for the conflict graph of the SOS2(N_i) constraint. Furthermore, define*

$$\begin{aligned} A^{\eta+1,u} &= V \cap (u + 3\mathbb{Z}^{\eta}) \\ B^{\eta+1,u} &= \bigcup_{w \in V \cap (u+3\mathbb{Z}^{\eta})} \left\{ w + v \;\middle|\; v \in \{-1, 1\}^{\eta},\ \{w, w+v\} \in \bar{E} \right\} \end{aligned}$$

for each $u \in \{0,1,2\}^{\eta}$. Then $\left(\bigcup_{i=1}^{\eta} \{(A^{i,j}, B^{i,j})\}_{j=1}^{t_i}\right) \cup \{(A^{\eta+1,u}, B^{\eta+1,u})\}_{u \in \{0,1,2\}^2}$ is a biclique cover for $G^c_{\mathcal{T}}$, where

$$\begin{aligned} A^{i,j} &= \left(\prod_{\ell=1}^{i-1} [\![N_i]\!] \right) \times \tilde{A}^{i,j} \times \left(\prod_{\ell=i+1}^{\eta} [\![N_i]\!] \right) \\ B^{i,j} &= \left(\prod_{\ell=1}^{i-1} [\![N_i]\!] \right) \times \tilde{B}^{i,j} \times \left(\prod_{\ell=i+1}^{\eta} [\![N_i]\!] \right) \end{aligned}$$

for each $i \in [\![\eta]\!]$ and $j \in [\![t_1]\!]$.

In particular, using the Gray code construction as in Theorem 4 for each SOS2(N_i) constraint yields a biclique cover of $G^c_{\mathcal{T}}$ of depth $\sum_{i=1}^{\eta} \lceil \log_2(d_i) \rceil + 3^{\eta}$.

2.8.3 An even smaller formulation for bivariate grid triangulations

We are now in position to present a second biclique cover for arbitrary grid triangulations that is smaller than the one presented in Theorem 4, but more complex. We refer to this formulation as the *6-stencil* formulation, and return to it in Chapter 4.2.2, where we will see that it offers substantial computational improvements over existing formulations for arbitrary grid triangulations.

Theorem 6. *Consider an arbitrary grid triangulation on the grid $V = [\![N_1]\!] \times [\![N_2]\!]$, where $N_1 \equiv d_1 + 1$ and $N_2 \equiv d_2 + 1$. There exists an independent branching formulation of depth $\lceil \log_2(d_1) \rceil + \lceil \log_2(d_2) \rceil + 6$.*

To prove the result, we describe the construction in detail. First, we adopt the $\lceil \log_2(d_1) \rceil + \lceil \log_2(d_2) \rceil$ levels as defined in (2.16a-2.16d). These levels cover all edges in the conflict graph that are "far apart." To accomplish the triangle selection, we construct a biclique representation $\{(A^{3,k}, B^{3,k})\}_{k=1}^{6}$ to cover all "nearby" edges $\bar{E}^N = \{\, \{u,v\} \in [V]^2 \,\|\, |u-v||_\infty = 1 \,\}$ by applying a "stencil" construction along diagonal and anti-diagonal lines. Appropriately, we call the resulting independent branching representation the *6-stencil*, and we illustrate the construction in Figure 2-8.

For each $\rho \in \mathbb{Z}$, consider the diagonal and anti-diagonal line on the grid V, offset by ρ as

$$
\begin{aligned}
DL_\rho &\stackrel{\text{def}}{=} ((j, j+\rho) \in V : j \in \mathbb{N}) \\
ADL_\rho &\stackrel{\text{def}}{=} ((j, d_2 + 2 - j + \rho) \in V : j \in \mathbb{N}),
\end{aligned}
$$

with the ordering of the elements given as the first component increases (i.e. $DL_0 = ((1,1),(2,2),\ldots,(\min\{d_1+1, d_2+1\}, \min\{d_1+1, d_2+1\}))$. Take those nearby edges for which both ends lie on the diagonal line DL_ρ as $\bar{E}^{DL_\rho} = \left\{\, \{u,v\} \in \bar{E}^N \,\middle|\, u,v \in DL_\rho \,\right\}$, and analogously with $\bar{E}^{ADL_\rho} = \left\{\, \{u,v\} \in \bar{E}^N \,\middle|\, u,v \in ADL_\rho \,\right\}$ for the anti-diagonal lines. We can observe that $\bar{E}^N = (\bigcup_{\rho \in \mathbb{Z}} \bar{E}^{DL_\rho}) \cup (\bigcup_{\rho \in \mathbb{Z}} \bar{E}^{ADL_\rho})$.

Fix some $\rho \in \mathbb{Z}$, and focus for the moment on the diagonal line DL_ρ, which we presume is nonempty (else take $\tilde{A}^{DL,\rho} = \tilde{B}^{DL,\rho} = \varnothing$ and proceed). Take $(u_1, \ldots, u_\zeta)$ as the ordering of the subset $\Phi_\rho = \bigcup \{\{u,v\} \in \bar{E}^{DL_\rho}\} \subseteq DL_\rho$ of the breakpoints on the diagonal line incident to edges in $\bar{E}^N$; it inherits its ordering from the ordering of DL_ρ. We will take $\tilde{A}^{DL,\rho}, \tilde{B}^{DL,\rho} \subset V$ as a partition of Φ_ρ (i.e. $\tilde{A}^{DL,\rho} \cup \tilde{B}^{DL,\rho} = \Phi_\rho$ and $\tilde{A}^{DL,\rho} \cap \tilde{B}^{DL,\rho} = \varnothing$) in the following way: we place $u^1 \in \tilde{A}^{DL,\rho}$, then $u^2 \in \tilde{B}^{DL,\rho}$ if $\{u^1, u^2\} \in \bar{E}^N$, and otherwise $u^2 \in \tilde{A}^{DL,i}$. We repeat this procedure for $k = 2, 3, \ldots, \zeta$, alternating the sets we place subsequent elements in (i.e. $\{u^{k-1}, u^k\} \in \tilde{A}^{DL,\rho} * \tilde{B}^{DL,\rho}$) if and only if the pair corresponds to a "nearby edge" (i.e. $\{u^{k-1}, u^k\} \in \bar{E}^N$); otherwise, we place the subsequent element in the same set as the previous one (i.e. either $\{u^{k-1}, u^k\} \in \tilde{A}^{DL,\rho}$ or $\{u^{k-1}, u^k\} \in \tilde{B}^{DL,\rho}$). Intuitively, this means that if there is a "gap" in $\bar{E}^N$ along the diagonal line, we ensure that both ends of the gap lie in the same side of the biclique, to avoid adding an edge that does not appear in $\bar{E}$ and ensure we satisfy condition 3. As a concrete example, refer to the first panel in Figure 2-8. For $\rho = 3$, we have

$$\tilde{A}^{DL,3} = \{(1,4),(4,7),(5,8)\} \qquad \text{and} \qquad \tilde{B}^{DL,3} = \{(2,5),(3,6),(6,9)\},$$

whereas for $\rho = -3$ we have

$$\tilde{A}^{DL,-3} = \{(5,2),(8,5)\} \qquad \text{and} \qquad \tilde{B}^{DL,-3} = \{(6,3),(7,4)\}.$$

After applying an analogous construction to the anti-diagonal edges to produce $\{(\tilde{A}^{ADL,\rho}, \tilde{B}^{ADL,\rho})\}_{\rho\in\mathbb{Z}}$, we have constructed the requisite bicliques to satisfy conditions 2 and 3:

$$\bar{E}^N \subseteq \left(\bigcup_{\rho\in\mathbb{Z}} (\tilde{A}^{DL,\rho} * \tilde{B}^{DL,\rho})\right) \cup \left(\bigcup_{\rho\in\mathbb{Z}} (\tilde{A}^{ADL,\rho} * \tilde{B}^{ADL,\rho})\right) \subseteq \bar{E}.$$

It just remains to show that we can aggregate these (infinitely many) bicliques into just 6 levels, while maintaining the inclusion in the edge set $\bar{E}$. For this, note that for any $\rho, \kappa \in \mathbb{Z}$ with $|\rho - \kappa| \geqslant 3$, we have that $||u - v||_\infty \geqslant 2$ for each $u \in DL_\rho$ and $v \in DL_\kappa$. Furthermore, $\{u, v\} \in \bar{E}\backslash\bar{E}^N$ for any such $u, v \in V$ such that $||u - v||_\infty \geqslant 2$. Therefore, for any $a \in \tilde{A}^{DL,\rho} \subseteq DL_\rho$ and $v \in \tilde{B}^{DL,\kappa} \subseteq DL_j$, we have that $\{u, v\} \in \bar{E}$ necessarily. This holds analogously for anti-diagonal lines, so if we define

$$A^{DL,\alpha} = \bigcup_{\rho\in(3\mathbb{Z}+\alpha)} \bar{A}^{DL,\rho}, \qquad B^{DL,\alpha} = \bigcup_{\rho\in(3\mathbb{Z}+\alpha)} \bar{B}^{DL,\rho}$$
$$A^{ADL,\alpha} = \bigcup_{\rho\in(3\mathbb{Z}+\alpha)} \bar{A}^{ADL,\rho}, \qquad B^{ADL,\alpha} = \bigcup_{\rho\in(3\mathbb{Z}+\alpha)} \bar{B}^{ADL,\rho}$$

for each $\alpha \in \{0, 1, 2\}$ we have that

$$\bar{E}^N \subseteq \left(\bigcup_{\alpha\in\{0,1,2\}} A^{DL,\alpha} * B^{DL,\alpha})\right) \cup \left(\bigcup_{\alpha\in\{0,1,2\}} A^{ADL,\alpha} * B^{ADL,\alpha})\right) \subseteq \bar{E},$$

thus completing the construction.

2.8.4 The SOSk constraint

In this subsection, we will see how we may use the graph union construction to produce an IB scheme for SOS$k(N)$ of depth $\log_2(N/k) + \mathcal{O}(k)$, for any k and N. Similar to the construction for grid triangulations, we first construct an initial family of bicliques based on the SOS2 constraint. Next, we expand this onto a larger node set by the graph product construction. Finally, we complete the biclique cover by combining a family of sufficiently separated stars. For grid triangulations, this approach meant applying SOS2 constraints horizontally and vertically, and taking a graph product of the two. One way to interpret this is as an SOS2 constraint applied to groups of aggregated nodes in the ground set (e.g. when SOS2 is applied horizontally, we group all elements with the same horizontal coordinate into a single group). For the SOS$k(N)$ constraint, we will apply the SOS2 constraint to the groups obtained by partitioning the N original ground elements into $\lceil N/k \rceil$ subsets of k consecutive elements. The following simple lemma shows how this grouping can also be represented through a graph product. For the remainder of the section, we assume that N/k is integer; if this is not true, we artificially introduce $\lceil N/k \rceil k - N$ nodes such that this is the case, construct the formulation in Theorem 7, and remove the artificial nodes from the formulation afterwards.

Lemma 7. *Let $V = [\![N]\!]$, $k \leqslant N$, $\mathcal{T} \equiv \mathcal{T}^{\text{SOS}}_{N,k}$ correspond to the SOS$k(N)$ constraint, and $G^c_{\mathcal{T}} = (V, \bar{E})$ be the corresponding conflict graph. Let $G^1 = ([\![N/k]\!], E^1)$ be the conflict graph for SOS2(N/k) and $G^2 = ([\![0, k-1]\!], \varnothing)$ be the empty graph on k nodes. Then $G^1 \times G^2$ is isomorphic to a subgraph $\hat{G} = (V, \hat{E})$ of $G^c_{\mathcal{T}}$ wherein $\hat{E} \subseteq \bar{E}$, and each edge $\{u, v\} \in \bar{E}$ with $|u - v| \geqslant 2k$ is contained $\{u, v\} \in \hat{E}$.*

Proof. Let $G^1 \times G^2 = (V', E')$. Consider the bijection $f : V \to V'$ given by $f(u) = (\text{div}(u, k), \text{mod}(u, k))$, where $\text{div}(u, k) \overset{\text{def}}{=} \lfloor u/k \rfloor$ and $\text{mod}(u, k) \overset{\text{def}}{=} u - k\,\text{div}(u, k)$ are the quotient and remainder of the division of u by k, so that $f^{-1}(m, r) = km + r$. We have that $\{(m, r), (m', r')\} \in E'$ if and only if $\{m, m'\} \in E^1$, which in turn is equivalent

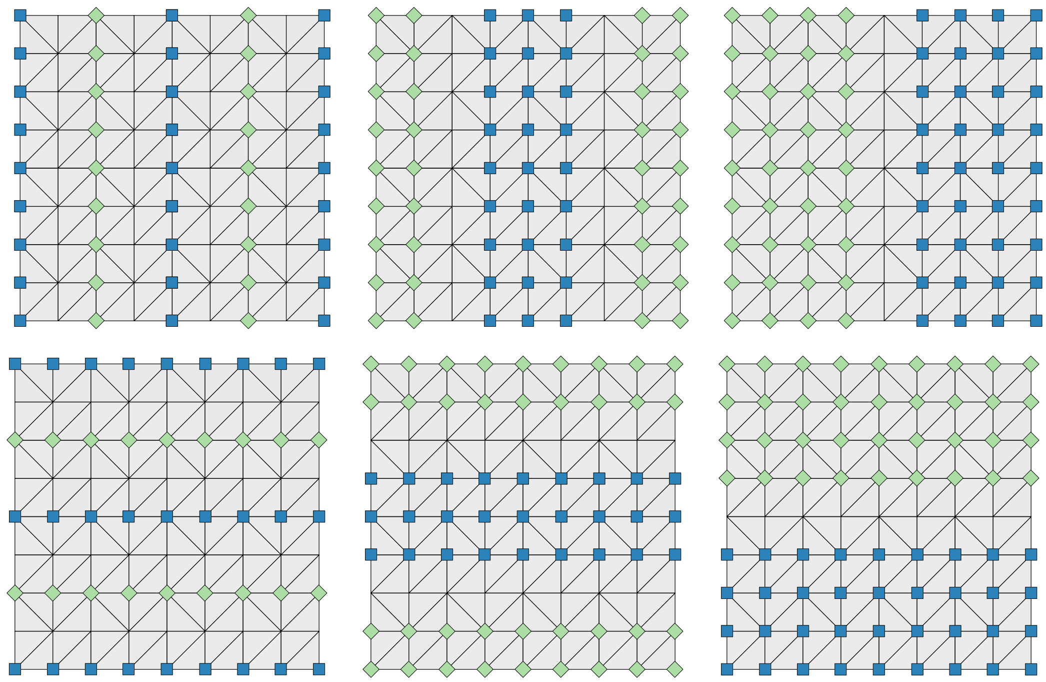


Figure 2-7: The aggregated SOS2 independent branching formulation for subrectangle selection. The sets $A^{1,k}$ (resp. $B^{1,k}$) are the squares (resp. diamonds) in the first row; similarly for the sets $A^{2,k}$ and $B^{2,k}$ in the second row.

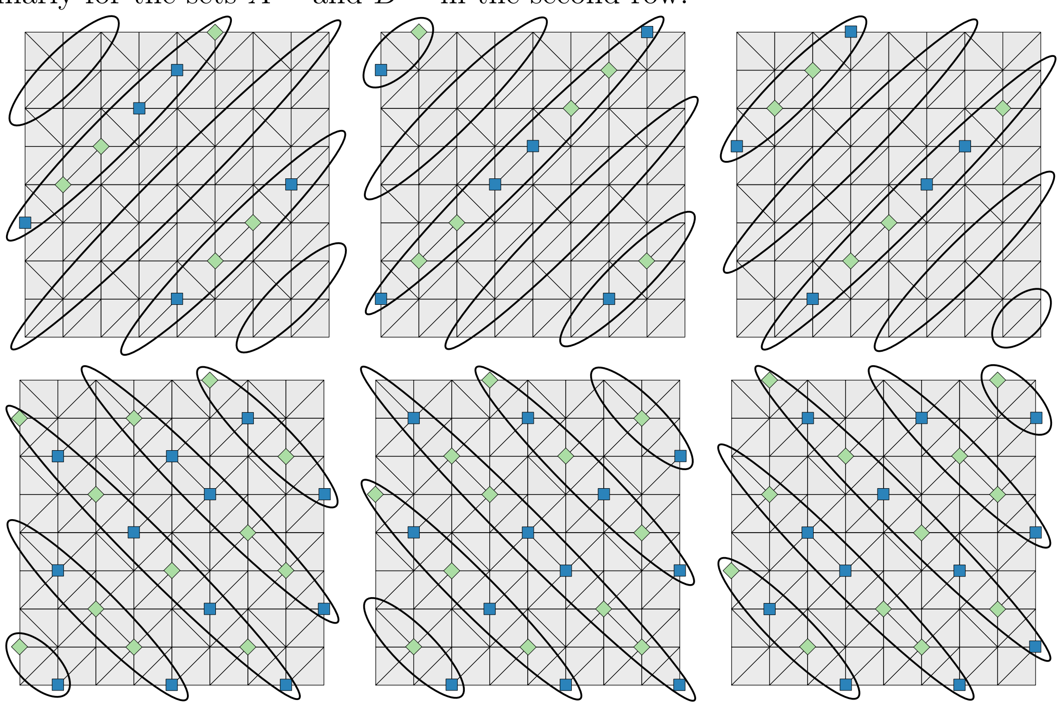


Figure 2-8: The 6-stencil triangle selection independent branching formulation. The sets $A^{DL,\alpha}$ (resp. $B^{DL,\alpha}$) are the squares (resp. diamonds) in the first row; similarly for the sets $A^{ADL,\alpha}$ and $B^{ADL,\alpha}$ in the second row. The diagonal/antidiagonal lines considered in each level are circled.

to $|m - m'| \geqslant 2$. Therefore, for any $\{(m, r), (m', r')\} \in E'$, we have

$$\begin{aligned}\left|f^{-1}(m, r) - f^{-1}(m', r')\right| &= |(km + r) - (km' + r')| \\ &= |k(m - m') + (r - r')| \\ &\geqslant k|m - m'| + |r - r'| \\ &\geqslant 2k,\end{aligned}$$

and hence $\{f^{-1}(m, r), f^{-1}(m', r')\} \in \bar{E}$, i.e. $\hat{E} \subseteq \bar{E}$. For the second condition, see that if $u, v \in V$ are such that $|u - v| \geqslant 2k$, then $|\operatorname{div}(u) - \operatorname{div}(v)| \geqslant 2$, and therefore $\{f(u), f(v)\} \in E'$. $\square$

We can then cover the remaining edges with the following bicliques obtained by stitching together families of sufficiently separated stars.

Corollary 6. *Let* $V = [\![N]\!]$, $k \leqslant N$, $\mathcal{T} \equiv \mathcal{T}^{\mathrm{SOS}}_{N,k}$ *correspond to the SOS*$k(N)$ *constraint, and* $G^c_{\mathcal{T}}$ *be the corresponding conflict graph. For all* $w \in V$*, define* $A(w) \overset{\mathsf{def}}{=} \{w\}$ *and* $B(w) \overset{\mathsf{def}}{=} \{\, u \in V \mid k \leqslant |u - w| < 2k \,\}$*. Then*

$$\left(\bigcup_{w \in V \cap (u + 3k\mathbb{Z})} A(w), \bigcup_{w \in V \cap (u + 3k\mathbb{Z})} B(w) \right)$$

is a biclique of $G^c_{\mathcal{T}}$ *for any* $u \in V$*.*

Proof. Direct from Lemma 6 by considering the family of bicliques

$$\{(A(w), B(w)\}_{w \in V \cap (u + 3k\mathbb{Z})}$$

and noting that, for distinct $u, v \in V \cap (u + 3k\mathbb{Z})$, $|u - v| \geqslant 3k$, and so $(A(u), B(v))$ is also a biclique for $G^c_{\mathcal{T}}$. $\square$

Finally, we can combine both classes of bicliques with Lemma 5 to construct a complete biclique cover for SOS$k(N)$. See Figure 2-9 for an example of the resulting construction.

Theorem 7. *Let $V = [\![N]\!]$, $k \leqslant N$, $\mathcal{T} \equiv \mathcal{T}^{\mathrm{SOS}}_{N,k}$ correspond to the SOSk(N) constraint on V, and $G^c_{\mathcal{T}}$ be the corresponding conflict graph. Let $\{(\tilde{A}^{1,j}, \tilde{B}^{1,j})\}_{j=1}^{t_1}$ be a biclique cover for the conflict graph of the SOS2(N/k) constraint, and take*

$$A^{2,j'} = \bigcup_{i=0}^{\lceil \frac{N}{3k} \rceil} \{\, \tau \in V \mid \tau = j' + (3i-3)k \,\}$$

$$B^{2,j'} = \bigcup_{i=0}^{\lceil \frac{N}{3k} \rceil} \{\, \tau \in V \mid j' + (3i-2)k \leqslant \tau \leqslant j' + (3i-1)k \,\}$$

for all $j' \in [\![3k]\!]$. Then $\{(A^{1,j}, B^{1,j})\}_{j=1}^{t} \cup \{(A^{2,j'}, B^{2,j'})\}_{j'=1}^{[\![3k]\!]}$ is a biclique cover for $G^c_{\mathcal{T}}$, where

$$A^{1,j} = \left\{ \tau \in V \;\middle|\; \lceil \tau/k \rceil \in \tilde{A}^{1,j} \right\}, \qquad B^{1,j} = \left\{ \tau \in V \;\middle|\; \lceil \tau/k \rceil \in \tilde{B}^{1,j} \right\},$$

for each $j \in [\![t]\!]$.

In particular, if $\{h^i\}_{i=1}^{\lceil N/k \rceil - 1} \subseteq \{0,1\}^{\lceil \log_2(\lceil N/k \rceil - 1) \rceil}$ is a Gray code where $h^0 \stackrel{\mathsf{def}}{=} h^1$ and $h^{\lceil N/k \rceil} \stackrel{\mathsf{def}}{=} h^{\lceil N/k \rceil - 1}$, then a biclique cover of $G^c_{\mathcal{T}}$ of depth $\lceil \log_2(\lceil N/k \rceil - 1) \rceil + 3k$ is given by $\{(A^{1,j}, B^{1,j})\}_{j=1}^{\lceil \log_2(\lceil N/k \rceil - 1}} \cup \{(A^{2,j'}, B^{2,j'})\}_{j'=1}^{[\![3k]\!]}$, where

$$A^{1,j} = \left\{ \tau \in V : h_j^{\lceil \tau/k \rceil - 1} = h_j^{\lceil \tau/k \rceil} = 0 \right\}, \qquad B^{1,j} = \left\{ \tau \in V : h_j^{\lceil \tau/k \rceil - 1} = h_j^{\lceil \tau/k \rceil} = 1 \right\}$$

for all $j \in [\![\lceil \log_2(\lceil N/k \rceil - 1) \rceil]\!]$.

Proof. Take $G^1 \stackrel{\mathsf{def}}{=} ([\![N/k]\!], E^1)$ as the conflict graph for SOS2(N/k), $G^2 \stackrel{\mathsf{def}}{=} ([\![0, k-1]\!], \varnothing)$ as the empty graph on k nodes, and $G^3 \stackrel{\mathsf{def}}{=} \left(V, \bigcup_{j'=1}^{3k} A^{2,j'} * B^{2,j'}\right)$. Let $\hat{G}$ be the subgraph of $G^c_{\mathcal{T}}$ from Lemma 7, which is isomorphic to $G^1 \times G^2$ through the bijection $g : [\![N/k]\!] \times [\![0, k-1]\!] \to V$ with $g(m, r) = km + r$. Then we have that $G^c_{\mathcal{T}} = \hat{G} \cup G^3$, after applying Lemma 7 and using the fact that the edges of G^3 contain the edges of $G^c_{\mathcal{T}}$ not included in $\hat{G}$. The result then follows from Lemma 4, Lemma 5, Lemma 7, and Corollary 5. □

We note that, when $k = \mathcal{O}(\log(N))$, this biclique cover yields a binary MIP formulation that is asymptotically tight (with respect to the number of binary variables)

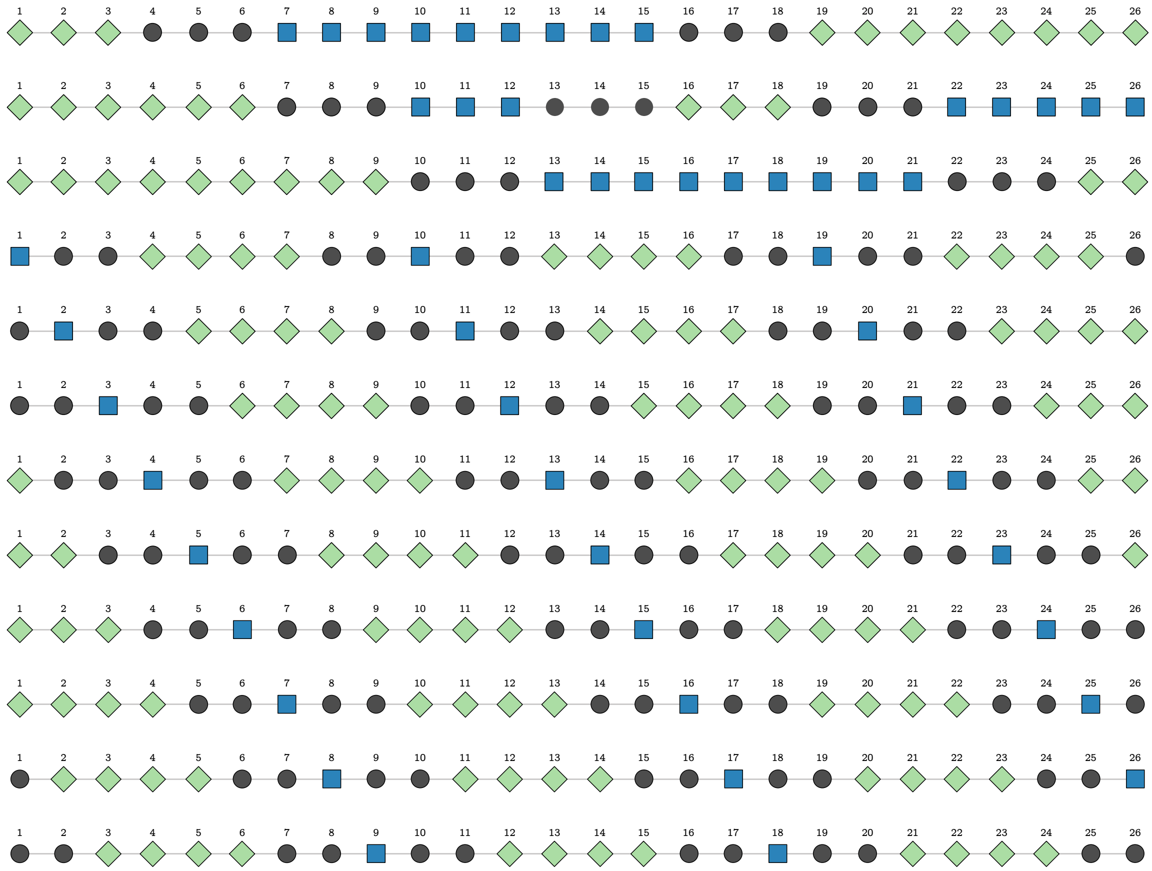


Figure 2-9: Visualizations of the biclique cover from the proof of Theorem 7 for SOS3(26). Each row corresponds to some level j, and the sets A^j and B^j are the blue squares and green diamonds, respectively. The first three rows correspond the the "first stage" of the biclique cover $\{(A^{1,j}, B^{1,j})\}_{j=1}^{3}$, and the second nine correspond to the "second stage" $\{(A^{2,j}, B^{2,j})\}_{j=1}^{9}$.

with our lower bound of $\lceil \log_2(N-k+1) \rceil$ from Proposition 1. We can also show an absolute lower bound of depth k for any biclique cover for SOSk. This implies that when $k = \omega(\log(N))$, although the formulation from Theorem 7 is not tight with respect to the lower bound from Proposition 1, it is asymptotically the smallest possible formulation in the pairwise IB framework.

Proposition 8. *Any biclique cover for the conflict graph of $G^c_{\mathcal{T}}$ of the SOSk(N) constraint with $\mathcal{T} \equiv \mathcal{T}^{\text{SOS}}_{N,k}$ must have depth at least* $\min\{k, N-k\}$.

Proof. Define $\gamma \stackrel{\text{def}}{=} \min\{k, N-k\}$ and consider any possible biclique cover $\{(A^j, B^j)\}_{j=1}^t$ for the conflict graph $G^c_{\mathcal{T}} = (V, \bar{E})$. The biclique cover must separate the edges $\{(\tau, \tau+k)\}_{\tau=1}^{\gamma}$. Consider a level j of the biclique cover that contains edge $\{\tau, \tau+k\}$ for some $\tau \in [\![\gamma]\!]$; w.l.o.g., $\tau \in A^j$ and $\tau + k \in B^j$. Consider the possibility that the same level j separates another such edge in the set, e.g. $(\tau', \tau'+k)$ for $\tau' \in [\![\gamma]\!]$, where w.l.o.g. $\tau < \tau'$. That would imply that either $\tau' \in A^j$ or $\tau' \in B^j$. In the case that $\tau' \in A^j$, we have that $\bar{E}^j$ contains the edge $\{\tau', \tau+k\}$. However, since $|(\tau+k)-\tau'| = \tau+k-\tau' < \tau+k-\tau = k$, this implies that the biclique cover separates a feasible edge, a contradiction. In the case where $\tau' \in B^j$, we have that $\bar{E}^j$ contains the edge $\{\tau, \tau'\}$, and as $\tau' - \tau < k \leqslant \gamma$ from the definition of our set of edges, a similar argument holds. Therefore, each edge $\{\{\tau, \tau+k\}\}_{\tau=1}^{\gamma}$ must be uniquely contained in some level of the biclique cover, giving the result. □

Furthermore, when $k = \lfloor N/2 \rfloor$, this proposition gives a lower bound on the depth of a biclique cover that is asymptotically tight with the upper bound of N from Proposition 6. In other words, in this particular regime, we have that the SOSk constraint admits a pairwise IB-based formulation, but only one that is relatively large ($\Omega(|\mathcal{T}|) = \Omega(|V|)$ binary variables and constraints).

Chapter 3

Building formulations geometrically: Embeddings.

In the previous chapter, we explored the strengths and limitations of a combinatorial approach for building MIP formulations. In this chapter, we will investigate a geometric approach to the same task. Our approach will allow us to readily study general integer MIP formulations, and we present a generic MIP formulation construction technique that is geometric in nature. We will see that, in terms of formulation size, there are relatively limited gains to be had from general integer (as opposed to binary) MIP formulations. However, our new approach will allow to build small, strong MIP formulations for univariate piecewise linear functions with desirable branching behavior in such a way that is (to the best of our knowledge) only attainable with general integer variables. In the following chapter, we will study these new formulations computationally and see that they improve on the state-of-the-art logarithmic formulation of Vielma et al. [133, 135].

3.1 The embedding approach

We will construct formulations for disjunctive sets $D = \bigcup_{i=1}^{d} P^i$ through what is known as the embedding approach [131]. We assign each alternative P^i a unique *code* $h^i \in \mathbb{Z}^r$. We call such a collection of distinct vectors $H = (h^i)_{i=1}^{d}$ an *encoding*. Given the family of polyhedra $\mathcal{P} = (P^i)_{i=1}^d$ and the encoding H, we construct the *embedding* of D in a higher-dimensional space as

$$\operatorname{Em}(\mathcal{P}, H) \stackrel{\mathsf{def}}{=} \bigcup_{i=1}^{d} (P^i \times \{h^i\}).$$

This object is useful as projecting out the integer variables gives us the original disjunctive set: $\operatorname{Proj}_x(\operatorname{Em}(\mathcal{P}, H)) = D$. Moreover, if the encoding satisfies a natural geometric condition, then its convex hull $Q(\mathcal{P}, H) \stackrel{\mathsf{def}}{=} \operatorname{Conv}(\operatorname{Em}(\mathcal{P}, H))$ is the LP relaxation of a non-extended ideal formulation for D.

Definition 9. *A set $H \subset \mathbb{R}^r$ is:*

- in convex position *if* $\operatorname{ext}(\operatorname{Conv}(H)) = H$*.*
- hole-free *if* $\operatorname{Conv}(H) \cap \mathbb{Z}^r = H$*.*

Proposition 9. *Take a family of bounded polyhedron $\mathcal{P} = (P^i)_{i=1}^d$. If $H = (h^i)_{i=1}^d$ is a hole-free encoding in convex position, then $\{ (x, z) \in Q(\mathcal{P}, H) \mid z \in \mathbb{Z}^r \}$ is an ideal formulation for $\bigcup_{i=1}^d P^i$.*

Proof. The validity of the formulation will follow from Corollary 8 to come. The ideal property follows from the definition of the embedding object, as

$$\operatorname{ext}(\operatorname{Conv}(\operatorname{Em}(\mathcal{P}, H))) \subseteq \operatorname{Em}(\mathcal{P}, H) \subset \mathbb{R}^n \times \mathbb{Z}^r.$$

□

We refer to the resulting formulation, given by the LP relaxation $Q(\mathcal{P}, H)$, as the *embedding formulation* for the particular choice of $\mathcal{P}$ and H. The condition of Proposition 9 is sufficient, but not necessary, as we will investigate further in this thesis.

An immediate useful corollary of this result is that any binary encoding $H \subseteq \{0,1\}^r$ will lead to a valid binary MIP formulation.

Corollary 7. *Take a family of bounded polyhedron* $\mathcal{P} = (P^i)_{i=1}^d$. *If* $H = (h^i)_{i=1}^d \subseteq \{0,1\}^r$ *is a binary encoding, then* $\{ (x,z) \in Q(\mathcal{P}, H) \mid z \in \{0,1\}^r \}$ *is an ideal formulation for* $\bigcup_{i=1}^d P^i$. *Moreover, we must necessarily have* $r \geqslant \lceil \log_2(d) \rceil$.

Proof. The first follows from Proposition 9 as $\{0,1\}^r$ is hole-free and in convex position, and so any subset will be as well. The second follows as H must be comprised of d distinct point from the definition of an encoding. □

The embedding approach is universal in the sense that any bounded ideal formulation is equivalent to an embedding formulation (potentially after projecting out auxiliary continuous variables), as the following result shows. Given R, define $\text{Slice}(R; z) \stackrel{\text{def}}{=} \{ x \in \mathbb{R}^n \mid \exists w \text{ s.t. } (x, w, z) \in R \}$ and $Z(R) \stackrel{\text{def}}{=} \{ z \in \mathbb{Z}^r \mid \exists x, w \text{ s.t. } (x, w, z) \in R \}$.

Proposition 10. *Take a polyhedra* $R \subseteq \mathbb{R}^{n+p+r}$ *such that* $\{ (x,w,z) \in R \mid z \in \mathbb{Z}^r \}$ *is an ideal formulation for* $D \subset \mathbb{R}^n$. *Then* $\tilde{R} = \text{Proj}_{(x,z)}(R)$ *is also an ideal formulation for* D. *Moreover, if* $Z(R)$ *is finite, then* $\tilde{R} = Q(\mathcal{P}, H)$ *for some encoding* $H = (h^i)_{i=1}^d$ *comprising of an ordering of* $Z(R)$*, along with the family* $\mathcal{P} = (\text{Slice}(R; h^i))_{i=1}^d$.

Inspired by Proposition 10, we make the following simplifying assumption for the remainder.

Assumption 3. *Given an LP relaxation* R*, there are only finitely many feasible integer points:* $|Z(R)| < \infty$.

We note that this assumption is *not* without loss of generality: there exists set D for which the only MIP formulations that exist require an infinite number of feasible integer points (as a trivial example, take the set $D = \mathbb{Z}$). However, this will be sufficient for the case we consider in this thesis: when D is the finite union of bounded polyhedra.

3.2 How many integer variables do we need?

At this point we pause ask a simple question: How small can we make r, the number of integer variables in our formulation? It is well-known that the number of integer variables tends to have a substantial impact on the performance of a MIP formulation, as in the worst case the search tree will need to enumerate a number of nodes exponential in r. The formulations we have seen up to this point have had r at least logarithmic in the number of alternatives d (i.e. $r \geqslant \lceil \log_2(d) \rceil$). Indeed, we have seen in Proposition 1 that this lower bound must hold for any binary MIP formulation for a combinatorial disjunctive constraint, and in Corollary 7 that this bound holds for binary embedding formulations as well. However, we will see in this section that if we allow ourselves to use general integer MIP formulations, the situation can sometimes be substantially different.

To start, we see that if we select the representation for our disjunctive set (namely, the sets $\mathcal{P}$) in an degenerate way, then it may not be meaningful to seek lower bounds on formulation size through our embedding approach. We illustrate with a simple example.

Example 3. *Take the family of intervals $\mathcal{P} = (P^i = [i-1, i] \subset \mathbb{R})_{i=1}^d$. Recall that the definition of an embedding requires us to select an encoding $H = (h^i)_{i=1}^d \subseteq \mathbb{Z}^r$ with* distinct *elements h^i. If we attempt to construct a binary MIP embedding formulation, then $H \subseteq \{0,1\}^d$, and we must necessarily have $r \geqslant \lceil \log_2(d) \rceil$ to satisfy the distinctness condition. Even if we allow ourselves to use general integer variables, we still must have $r \geqslant 1$. However, we can observe that $\bigcup_{i=1}^d P^i = [0, d]$ is itself an interval, and so the set is convex and we can construct an LP formulation for the constraint (that is, we do not need any integer variables).*

In Example 3, our choice of the family of sets $\mathcal{P}$ was redundant in the following sense.

Definition 10. *A family $\mathcal{P} = (P^i)_{i=1}^d$ is* redundant *if there exists some family $\tilde{\mathcal{P}} = (\tilde{P}^i)_{i=1}^{d'}$ where $\bigcup_{i=1}^d P^i = \bigcup_{i=1}^{d'} \tilde{P}^i$ and either 1) $d' < d$, or 2) $P^i \subseteq \tilde{P}^i$ for each $i \in [\![d]\!]$, and there exists some $i \in [\![d]\!]$ where this inclusion is strict. It is* irredundant *otherwise.*

Although redundancy is an annoyance when trying to produce lower bounds on r, there are situations where redundancy actually helps. For example, there exist situations where a redundant representation can yield ideal formulations with strictly fewer inequality constraints than any possible irredundant representation (an example with bivariate grid triangulations appeared in a preprint version of [66]).

Proposition 11. *Take a family of sets $\mathcal{P} = (P^i)_{i=1}^d$, $D = \bigcup_{i=1}^d P^i$, and an encoding $H = (h^i)_{i=1}^d \subset \mathbb{R}^r$. Take a polyhedron $R \subseteq \mathbb{R}^{n+p+r}$. Then the following statements are true.*

- $D = \bigcup_{i=1}^d \operatorname{Slice}(\operatorname{Em}(\mathcal{P}, H); h^i)$.
- *If H is hole-free, and*

$$\operatorname{Slice}(R; h^i) = P^i \quad \forall i \in [\![d]\!], \tag{3.1}$$

  then $\{ (x, w, z) \in R \mid z \in \mathbb{Z}^r \}$ is a formulation for D. Furthermore, if $\mathcal{P}$ is irredundant, then this relationship is an equivalency.

- *If H is not hole-free, then $\{ (x, w, z) \in R \mid z \in \mathbb{Z}^r \}$ is a valid formulation for D only if for each $z \in Z(R) \backslash H$,*

$$\operatorname{Slice}(R; z) \subseteq D.$$

Proof. The first result is immediate from the definition of the embedding. The second follows as the hole-free property implies that $H = Z(R)$, and so

$$\begin{aligned}
\operatorname{Proj}_x(R \cap (\mathbb{R}^{n+p} \times \mathbb{Z}^r)) &= \bigcup_{z \in Z(R)} \operatorname{Slice}(R; z) \\
&= \bigcup_{i=1}^d \operatorname{Slice}(R; h^i) \\
&= \bigcup_{i=1}^d P^i \equiv D.
\end{aligned}$$

The equivalency portion follows as from validity of the formulation given by R. That is, $P^i \subseteq \operatorname{Slice}(R; h^i)$, and if it is the case that this inclusion holds strictly, this implies that $\tilde{\mathcal{P}} = (\mathrm{Slice}(R; h^i))_{i=1}^d$ is a strictly dominating representation, contradicting the irredundancy of $\mathcal{P}$. The third follows similarly, as if R yields a valid formulation, we must have

$$\begin{aligned}
\mathrm{Proj}_x(R \cap (\mathbb{R}^{n+p} \times \mathbb{Z}^r)) &= \bigcup_{z \in Z(R)} \mathrm{Slice}(R; z) \\
&= \left(\bigcup_{i=1}^d \mathrm{Slice}(R; h^i) \right) \cup \left(\bigcup_{z \in Z(R) \setminus H} \mathrm{Slice}(R; z) \right) \\
&\subseteq \bigcup_{i=1}^d P^i \equiv D.
\end{aligned}$$

□

The second statement above gives us a simple sufficient statement to verify if an embedding formulation will be valid. Note that it differs from Proposition 10 as it does not require that H is in convex position. In addition, the third statement above implies that our representation $\mathcal{P}$ is redundant. Therefore, we will make the following assumption for the remainder of the chapter (though in Chapter 5 we will consider the implications if we relax this assumption).

Assumption 4. *We assume that any encoding H is hole-free.*

It is true that when $H \subseteq \{0,1\}^r$ is a binary encoding, the slice condition (3.1) is satisfied for the embedding formulation, given by the LP relaxation $R \equiv Q(\mathcal{P}, H)$ (this will follow immediately from Theorem 8 to come). However, if we consider general integer encodings H, some care needs to be taken to ensure that the slice condition (3.1) is indeed satisfied, as the following example illustrates.

Example 4. *Consider the sets $P^1 = [0,1]$, $P^2 = [2,3]$, and $P^3 = [4,5]$, along with the two ordered families $\mathcal{P} = (P^1, P^2, P^3)$ and $\tilde{\mathcal{P}} = (P^2, P^1, P^3)$. First, for any binary encoding $H \subseteq \{0,1\}^r$ with $r \geqslant 2$, the corresponding relaxations $Q(\mathcal{P}, H)$ and $Q(\tilde{\mathcal{P}}, H)$ satisfy condition (3.1) and yield valid binary MIP formulations. Moreover, any binary MIP formulation for either $\mathrm{Em}(\mathcal{P}, H)$ or $\mathrm{Em}(\tilde{\mathcal{P}}, H)$ will require at least two integer variables. However, if we take $H = (1,2,3)$, then $Q(\mathcal{P}, H)$ yields a valid*

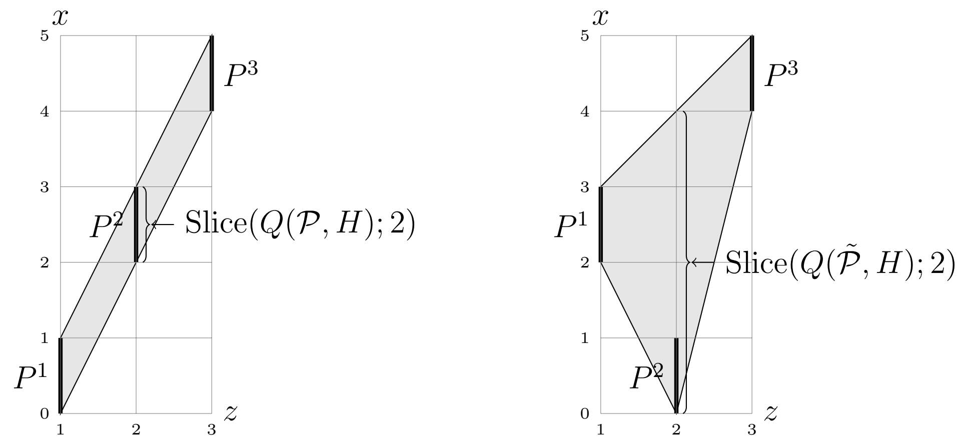


Figure 3-1: Two embeddings with the encoding $H = (1, 2, 3)$ and different orders of the sets as **(Left)** $\mathcal{P} = ([0, 1], [2, 3], [4, 5])$ and **(Right)** $\tilde{\mathcal{P}} = ([2, 3], [0, 1], [4, 5])$. The ordering $\mathcal{P}$ satisfies (3.1); the ordering $\tilde{\mathcal{P}}$ does not, as can be seen from the slice at $h = 2$.

MIP formulation with only one general integer variable. On the other hand, $Q(\tilde{\mathcal{P}}, H)$ does not satisfy condition (3.1), and indeed does not yield a valid MIP formulation. See Figure 3-1.

For more complex examples, in Figures 3-2 and 3-3 we embed two irredundant disjunctive constraints using different general integer embeddings with strictly fewer than $\lceil \log_2(d) \rceil$ integer variables. In Figure 3-2, we take the union of 8 non-overlapping intervals on the real line. When we embed the sets using a contiguous subset of the integers ($H = (k)_{k=1}^{8} \subseteq \mathbb{Z}$), the slice condition (3.1) is satisfied, and so $Q(\mathcal{P}, H)$ gives a valid MIP formulation. This is a valid formulation with only one integer variable, strictly less than the $\lceil \log_2(8) \rceil = 3$ lower bound for a binary encoding. Moreover, it is possible to generalize this construction to an arbitrary number of sets d that can be formulated using the simple one-dimensional encoding $H = (k)_{i=1}^{d} \subset \mathbb{Z}$.

In Figure 3-3, we consider a more complex disjunctive set of $d = 16$ points or intervals on the real line, embedded with a two-dimensional encoding. This gives a valid MIP formulation for the disjunctive set with 2 integer variables, as opposed to the $\log_2(16) = 4$ needed for a binary encoding.

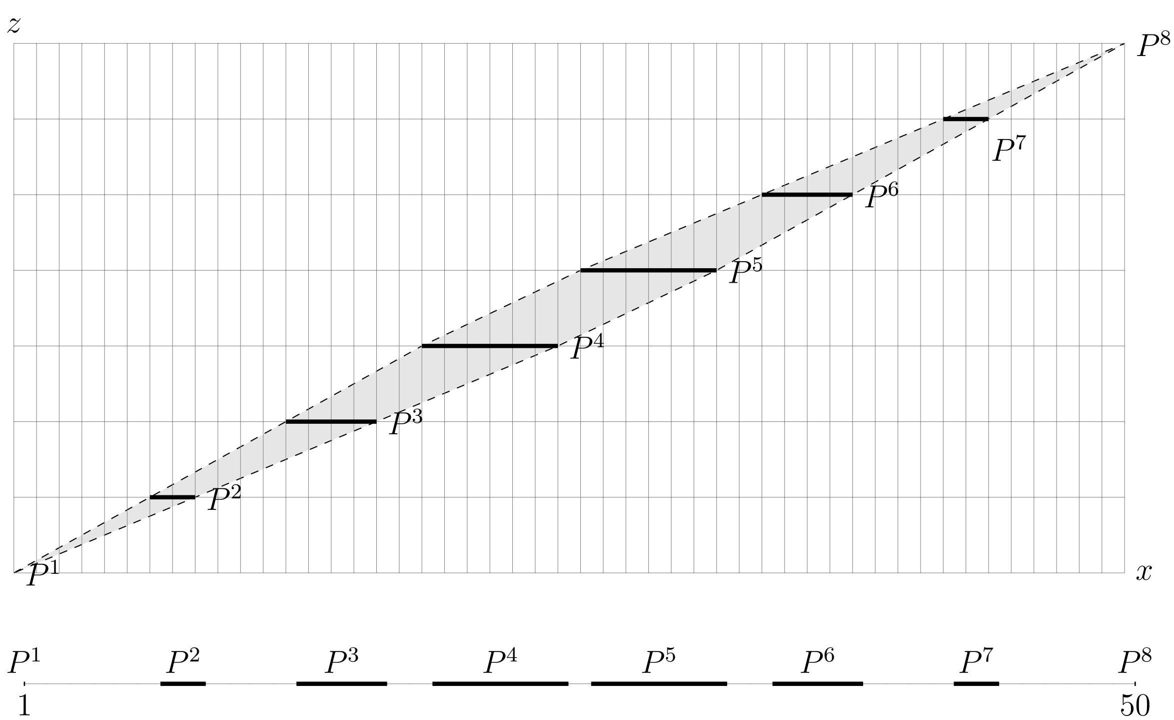


Figure 3-2: A family $\mathcal{P} = (P^i \subset \mathbb{R})_{i=1}^8$ for which the one-dimensional encoding $H = (k)_{k=1}^8 \subset \mathbb{Z}$ is such that $Q(\mathcal{P}, H)$ is a valid MIP formulation for $\bigcup_{i=1}^8 P^i$. **(Top)** A depiction of the embedding $\mathrm{Em}(\mathcal{P}, H)$ and its convex hull $Q(\mathcal{P}, H)$. **(Bottom)** The disjunctive set $\bigcup_{i=1}^8 P^i$ on the real line.

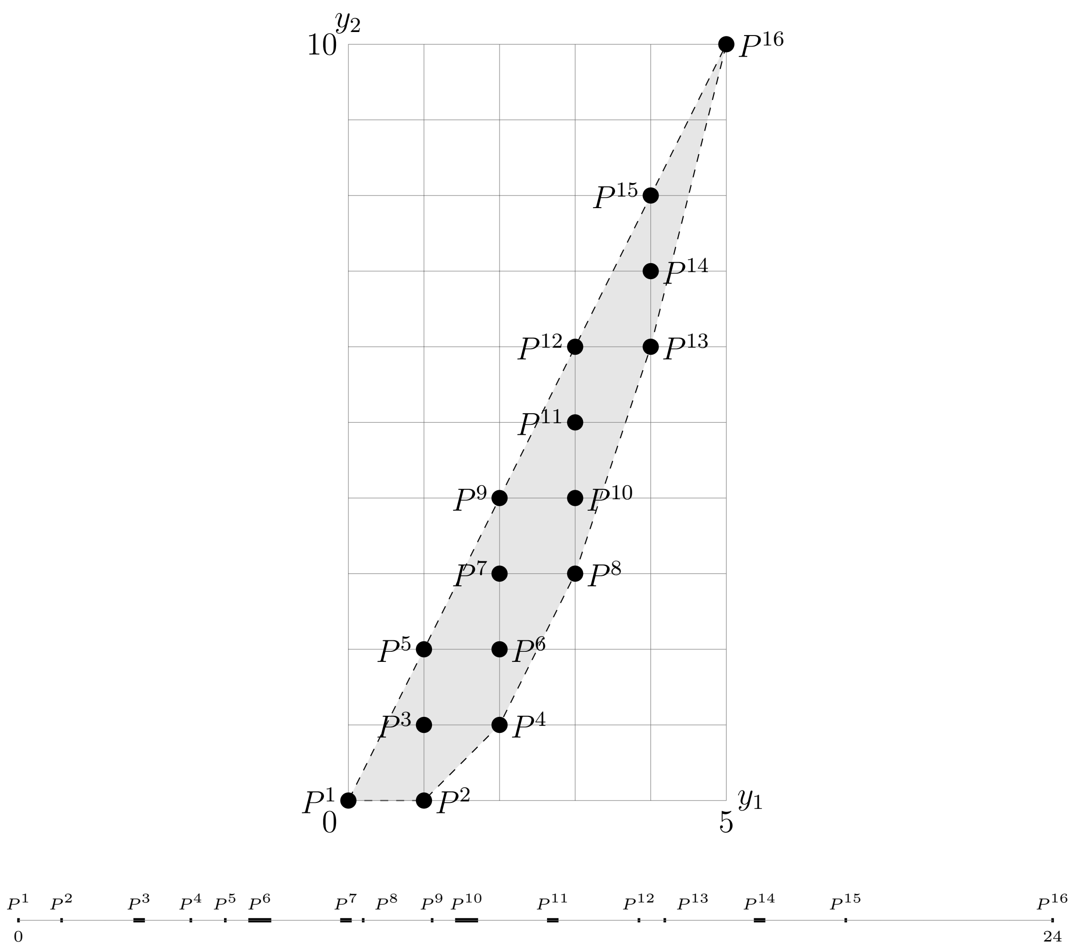


Figure 3-3: A family $\mathcal{P} = (P^i \subset \mathbb{R})_{i=1}^{16}$, along with an encoding $H = (h^i)_{i=1}^{16} \subseteq \mathbb{Z}^2$ of dimension 2. **(Top)** A depiction of H, the associated set P^i with each code h^i, and the convex hull $\text{Conv}(H)$ of the codes. **(Bottom)** The disjunctive set $\bigcup_{i=1}^{16} P^i$ on the real line.

3.2.1 A geometric characterization of when embeddings yield valid formulations

Example 4 and Figures 3-2 and 3-3 illustrate that it is possible to construct general integer MIP formulations for disjunctive sets with strictly fewer than $\lceil \log_2(d) \rceil$ integer variables, provided that the slices of the formulation are "well-behaved." At this point, we are prepared to present a geometric characterization of exactly when this is the case.

Theorem 8. *Take some irredundant family of bounded polyhedra $\mathcal{P} = (P^i)_{i=1}^d$ and a hole-free encoding $H = (h^i)_{i=1}^d \subset \mathbb{R}^r$. Then $F = \{ (x,z) \in Q(\mathcal{P},H) \mid z \in \mathbb{Z}^r \}$ is a valid formulation for $\bigcup_{i=1}^d P^i$ if and only if for each $i \in [\![d]\!]$, and for all $\lambda \in \Delta^d$ such that $h^i = \sum_{j=1}^d \lambda_j h^j$, it is the case that $P^i \supseteq \sum_{j=1}^d \lambda_j P^j$.*

Proof. To show the "only if" direction, take any $i \in [\![d]\!]$ and any $\lambda \in \Delta^d$ such that $h^i = \sum_{j=1}^d \lambda_j h^j$. Then

$$\begin{aligned}
\sum_{j=1}^d \lambda_j (P^j \times \{h^j\}) &= \sum_{j=1}^d (\lambda_j P^j \times \{h^i\}) \\
&\subseteq Q(\mathcal{P},H) \cap (\mathbb{R}^n \times \{h^i\}) \\
&= P^i \times \{h^i\},
\end{aligned}$$

where the inclusion follows from the definition $Q(\mathcal{P},H) \equiv \operatorname{Conv}(\operatorname{Em}(\mathcal{P},H))$, and the last equality follows from Proposition 11 and the slice condition (3.1). Taking the projection of both sides onto the x variables yields the set inclusion condition $P^i \supseteq \sum_{j=1}^d \lambda_j P^j$.

To show the "if" direction, by Proposition 11 it suffices to show that $P^i = \operatorname{Slice}(Q(\mathcal{P},H); h^i)$ for each $i \in [\![d]\!]$. Take any such $i \in [\![d]\!]$. From the definition of $\operatorname{Em}(\mathcal{P},H)$ and $Q(\mathcal{P},H)$, we immediately have that $P^i \subseteq \operatorname{Slice}(Q(\mathcal{P},H); h^i)$. To show the reverse inclusion, consider some element $\hat{x} \in \operatorname{Slice}(Q(\mathcal{P},H); h^i)$. From the definition of the slice, it follows that $(\hat{x}, h^i) \in Q(\mathcal{P},H)$. Therefore, it is possible to express $(\hat{x}, h^i)$ as a convex combination of the points in $\operatorname{Em}(\mathcal{P},H)$. Or, equivalently, there exists some $\lambda \in \Delta^d$ and some points $\tilde{x}^j \in P^j$ for each $j \in [\![d]\!]$ such that $(\hat{x}, h^i) = \sum_{j=1}^d \lambda_j (\tilde{x}^j, h^j)$. However, from assumption we have that, since $h^i = \sum_{j=1}^d \lambda_j h^j$, then $\sum_{j=1}^d \lambda_j \tilde{x}^j \in \sum_{j=1}^d \lambda_j P^j \subseteq P^i$, and so $\hat{x} \in P^i$, completing the proof. □

In particular, this characterization tells us that as long as an encoding is in convex position, we do not have to worry about how we assign sets to each code, in the sense that the resulting embedding formulation is valid.

Corollary 8. *Take an encoding $H = (h^i)_{i=1}^d \subseteq \mathbb{Z}^r$. If H is in convex position, then $Q(\mathcal{P}, H)$ satisfies condition (3.1) for any family of sets $\mathcal{P} = (P^i)_{i=1}^d$. Furthermore, if H is hole-free and in convex position, then $r \geqslant \lceil \log_2(d) \rceil$ necessarily.*

Proof. The first follows from Theorem 8 as, if H is in convex position, then for each $i \in [\![d]\!]$ the only value $\lambda \in \Delta^d$ wherein $h^i = \sum_{j=1}^d \lambda_j h^j$ is $\lambda = \mathbf{e}^i$. The second follows as $\mathrm{Conv}(H)$ has at most 2^r extreme points [29, Proposition 3], implying that H has at most $d = 2^r$ elements. □

3.2.2 Negative results for combinatorial disjunctive constraints

We can specialize this result for combinatorial disjunctive constraints. In particular, we can prove a more general version of the negative result of Proposition 1, which states that for combinatorial disjunctive constraints: 1) redundancy is simple to detect and remove, and that 2) if the constraint is irredundant, it requires an encoding in convex position that uses at least $r \geqslant \lceil \log_2(d) \rceil$ integer variables. In other words, we cannot hope to get lucky and produce MIP formulations for combinatorial disjunctive constraints with very few integer variables, as we were able to in Example 4 and Figures 3-2 and 3-3.

Proposition 12. *Let $\mathcal{T} = (T^i \subseteq V)_{i=1}^d$ be the family of sets defining a combinatorial disjunctive constraint associated with* $\mathrm{CDC}(\mathcal{T})$.

- *$\mathcal{P}(\mathcal{T})$ is redundant if and only if there is some $\{i, j\} \in [d]^2$ where $T^i \subseteq T^j$.*

- *If $\mathcal{P}$ is irredundant and H is hole-free, then $Q(\mathcal{P}(\mathcal{T}), H)$ is a valid formulation for* $\mathrm{CDC}(\mathcal{T})$ *if and only if H is in convex position. In particular, any MIP formulation for* $\mathrm{CDC}(\mathcal{T})$ *must have at least $\lceil \log_2(d) \rceil$ integer variables.*

Proof. We begin by proving the first part, pertaining to redundancy. The "if" direction is immediate. For the "only if" direction, presume that $\mathcal{P}$ is a redundant representation, and $\tilde{\mathcal{P}} = (\tilde{P}^i)_{i=1}^{d'}$ is an irredundant representation for the same constraint. For each $\lambda \in \mathrm{CDC}(\mathcal{T}) = \bigcup_{i=1}^d P(T^i)$, we have that $\mathbf{e}^v \in D$ for each $v \in \mathrm{supp}(\lambda)$. Therefore, we can presume w.l.o.g. that the sets in $\tilde{\mathcal{P}}$ are chosen maximally, and so therefore it is itself a combinatorial disjunctive constraint. That is, for each $i \in \llbracket d' \rrbracket$, there exists some $\tilde{T}^i \subseteq V$ such that $\tilde{P}^i = P(\tilde{T}^i)$.

As $\mathcal{P}$ and $\tilde{\mathcal{P}}$ represent the same constraint, and $\tilde{\mathcal{P}}$ is irredundant, for each $i \in \llbracket d' \rrbracket$ there must exist some $j_i \in \llbracket d \rrbracket$ such that $\tilde{T}^i = T^{j_i}$. Furthermore, each element in the set $I = \{j_i\}_{i=1}^{d'} \subsetneq \llbracket d \rrbracket$ is distinct, else $\tilde{\mathcal{P}}$ is redundant. Therefore, since $d' < d$, there must be some element $k \in \llbracket d \rrbracket \backslash I$. As $\bigcup_{i=1}^{d'} P(\tilde{T}^i) = \bigcup_{i=1}^{d'} P(T^{j_i}) = \bigcup_{i=1}^{d} P(T^i)$, and $k \notin I$, it follows that $P(T^k) \subseteq \bigcup_{i \in I} P(T^{j_i})$. Consider the point $\lambda = \frac{1}{|T^k|} \sum_{v \in T^k} \mathbf{e}^v$. As $\lambda \in \bigcup_{i \in I} P(T^{j_i})$, it follows that there is some $\ell \in I$ such that $\lambda \in P(T^\ell)$. This in turn implies that $\mathbf{e}^v \in P(T^\ell)$ for each $v \in T^k$. This means that $T^k \subseteq T^\ell$, completing the proof.

Now we prove the second part. The "if" direction follows directly from Corollary 8. For the "only if" direction, presume for contradiction that H is not in convex position, yet F is a valid formulation for D. In other words, there exists a code (w.l.o.g. h^1) where $h^1 = \sum_{i=1}^d \lambda_i h^i$ for some $\lambda \in \Delta^V$ where $\lambda_1 = 0$ and there exist at least two fractional components of λ; w.l.o.g., $0 < \lambda_2, \lambda_3 < 1$. From Theorem 8, we have that $P(T^1) \supseteq \sum_{i=2}^d \lambda_i P(T^i)$. Consider some $v \in T^2$. Since $\mathbf{e}^v \in P(T^2)$ and each point on the standard simplex is nonnegative, we must have that $\tilde{\lambda} \in P(T^1)$ for some $\tilde{\lambda} \in \Delta^V$ with $\tilde{\lambda}_v \geqslant \lambda_v > 0$. Therefore, we must have $v \in T^1$. Repeating this for each $v \in T^2$, we conclude that $T^1 \supseteq T^2$. However, this contradicts the irredundancy assumption, and so we must have that H is in convex position. $\square$

3.3 A geometric construction for ideal formulations of any combinatorial disjunctive constraint

Vielma [131] gives an explicit geometric description for $Q(\mathcal{P}(\mathcal{T}_d^{\text{SOS2}}), H)$, which gives an embedding formulation for the SOS2 constraint for any binary encoding H. In other words, this characterizes all non-extended ideal formulations for the SOS2 constraint. Motivated by the computational efficacy of the formulations you may construct using that result (including a close relative of the `LogIB` formulation we introduce in Chapter 3.4), we extend the characterization to any combinatorial disjunctive constraint. This result provides an explicit geometric construction for $Q(\mathcal{P}(\mathcal{T}), H)$ for any combinatorial disjunctive constraint given by the family $\mathcal{T}$, paired with any encoding H that is in convex position.

Theorem 9. *Take the family of sets $\mathcal{T} = (T^i \subseteq V)_{i=1}^d$. Let $H = (h^i)_{i=1}^d \subset \mathbb{R}^r$ be an encoding in convex position. Furthermore, let $\Upsilon = \{\, \{i,j\} \in [d]^2 \mid T^i \cap T^j \neq \emptyset \,\}$, and presume that Υ is path connected in the sense that the associated graph $G = (\llbracket d \rrbracket, \Upsilon)$ is connected. Take $C = \{c^{i,j} \overset{\mathsf{def}}{=} h^i - h^j\}_{\{i,j\}\in\Upsilon}$, and $\mathcal{L} = \operatorname{span}(C)$. Define $M(b;\mathcal{L}) \overset{\mathsf{def}}{=} \{\, y \in \mathcal{L} \mid b \cdot y = 0 \,\}$ to be the hyperplane in the linear space $\mathcal{L}$ induced by the direction $b \neq \mathbf{0}^r$. If $\{b^i\}_{i=1}^\Gamma \subset \mathbb{R}^r \backslash \{\mathbf{0}^r\}$ is such that $\{M(b^i;\mathcal{L})\}_{i=1}^\Gamma$ is the set of linear hyperplanes spanned by C in $\mathcal{L}$, then $(\lambda, z) \in Q(\mathcal{P}(\mathcal{T}), H)$ if and only if*

$$\sum_{v\in V} \lambda_v \min_{j:v\in T^j} \{b^i \cdot h^j\} \leqslant b^i \cdot z \leqslant \sum_{v\in V} \lambda_v \max_{j:v\in T^j} \{b^i \cdot h^j\} \quad \forall i \in \llbracket \Gamma \rrbracket \tag{3.2a}$$

$$(\lambda, z) \in \Delta^V \times \operatorname{aff}(H). \tag{3.2b}$$

Proof. For notational simplicity, we will presume that $V = \llbracket n \rrbracket$ for the proof. It is straightforward to show the "only if" direction. Take $B = \operatorname{ext}(Q(\mathcal{P}(\mathcal{T}), H))$ as the set of all extreme points. Note that, as we are working with a combinatorial disjunctive constraint, each extreme point will take the form $(\mathbf{e}^w, h^k) \in B$ for some $w \in T^k$. This shows that (3.2b) is trivially satisfied. Moreover, for each $(\hat{\lambda}, \hat{z}) \equiv (\mathbf{e}^w, h^k) \in B$ and each $i \in [\![\Gamma]\!]$, we have that

$$\sum_{v \in V} \hat{\lambda}_v \min_{j : v \in T^j} \{b^i \cdot h^j\} = \min_{j : w \in T^j} \{b^i \cdot h^j\} \leqslant b^i \cdot h^k \equiv b^i \cdot \hat{z},$$

where the inequality follows as $w \in T^k$. The other side of the inequality follows analogously, showing that each constraint in (3.2a) is also satisfied. Therefore, for the remainder we will focus on the "if" direction.

We start for the "if" direction by showing that $\mathcal{L} = \operatorname{aff}(H) - h^1$. For notational convenience, presume throughout that if we write an inclusion of the form $\{i, j\} \in \Upsilon$, it is implied that $i < j$, and so there is a well-defined ordering for the elements. To show that $\operatorname{span}(C) \subseteq \operatorname{aff}(H) - h^1$, take some $z \in \operatorname{span}(C)$, and so there exist multipliers $\gamma_{i,j}$ such that $z = \sum_{\{i,j\} \in \Upsilon} \gamma_{i,j}(h^i - h^j) = \sum_{i=1}^d \bar{\gamma}_i h^i$, where $\alpha_i = \sum_{j : \{i,j\} \in \Upsilon} \gamma_{i,j} - \sum_{k : \{k,i\} \in \Upsilon} \gamma_{k,i}$. Then $z = (\alpha_1 + 1)h^1 + (\sum_{i=2}^d \alpha_i h^i) - h^1$, i.e. $z \in \operatorname{aff}(H) - h^1$, as $(\alpha_1 + 1) + \sum_{i=2}^d \alpha_i = 1 + \sum_{i=1}^d \left(\sum_{j : \{i,j\} \in \Upsilon} \gamma_{i,j} - \sum_{k : \{k,i\} \in \Upsilon} \gamma_{k,i} \right) = 1 + 0$, and so they form affine multipliers. To show that $\operatorname{span}(C) \supseteq \operatorname{aff}(H) - h^1$, take some $z \in \operatorname{aff}(H) - h^1$, and so there exists multipliers μ_i such that $z = (\sum_{i=1}^d \mu_i h^i) - h^1$ and $\sum_{i=1}^d \mu_i = 1$. As G is connected, there exists some closed path $\{t_1 \equiv 1, t_2, \ldots, t_r, t_{r+1} \equiv 1\}$ on G that traverses each vertex. Take $\alpha_i \overset{\text{def}}{=} \frac{\mu_i}{\#\text{ of times path traverses } i}$ for each $i \in [\![d]\!]$. Then $z = \sum_{k=1}^r (h^{t_k} - h^{t_{k+1}}) \sum_{\ell=1}^r \alpha_{t_\ell} = \sum_{k=1}^r c^{t_k, t_{k+1}} \sum_{\ell=1}^r \alpha_{t_\ell}$ (using $\sum_{i=1}^d \mu_i = 1$ to show that the $\ell = r$ term in the sum produces the desired $-h^1$ term), and so therefore $z \in \operatorname{span}(C)$, as each $\{t_k, t_{k+1}\} \in \Upsilon$. This shows the result. Additionally, we note that the choice of h^1 to subtract from $\operatorname{aff}(H)$ was arbitrary.

Now, let F be a facet of $Q(\mathcal{P}(\mathcal{T}), H)$. By possibly adding or subtracting multiples of $\sum_{i=1}^n \lambda_i = 1$ and the equations defining $\operatorname{aff}(H)$, we may assume w.l.o.g. that F is induced by $\tilde{a} \cdot \lambda \leqslant \tilde{b} \cdot y$ for some $(\tilde{a}, \tilde{b}) \in \mathbb{R}^{n+r}$. We have that F is supported by some strict nonempty subset $\tilde{B} \subsetneq B$. Take $\tilde{\Upsilon} = \left\{ \{i, j\} \in \Upsilon \;\middle|\; \exists v \in [\![n]\!] \text{ s.t. } (\mathbf{e}^v, h^i), (\mathbf{e}^v, h^j) \in \tilde{B} \right\}$ and $\tilde{C} = \left\{ c^{i,j} \in C \;\middle|\; \{i, j\} \in \tilde{\Upsilon} \right\}$. In particular, we see that $\tilde{b} \cdot c^{i,j} = 0$ for each $c^{i,j} \in \tilde{C}$, as if $\{i, j\} \in \tilde{\Upsilon}$, this implies that there is some $v \in [\![n]\!]$ whereby $\tilde{a} \cdot \mathbf{e}^v = \tilde{b} \cdot h^i = \tilde{b} \cdot h^j$.

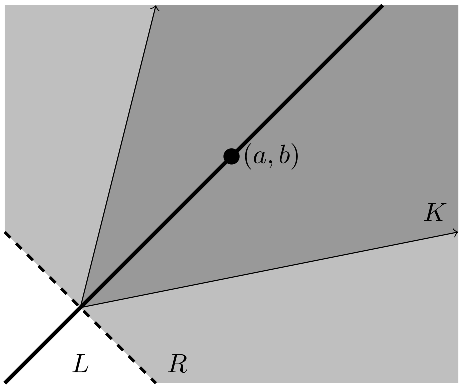
__**Case 1:** $\dim(\tilde{C}) = \dim(C)$__ In this case, we show that F corresponds to a variable bound on a single component of λ. As $\tilde{C} \subseteq C$ and $\dim(\tilde{C}) = \dim(C)$, we conclude that $\text{span}(\tilde{C}) = \text{span}(C) \equiv \mathcal{L}$. Then $\tilde{b} \in \mathcal{L}^{\perp}$, as $\tilde{b} \perp \tilde{C}$. Furthermore, $\mathcal{L}$ is the linear space parallel to $\text{aff}(H)$. Therefore, we can w.l.o.g. presume that $\tilde{b} = \mathbf{0}^r$, as (3.2b) constrains $z \in \text{aff}(H)$.

We observe that $\tilde{a} \neq \mathbf{0}^n$, as otherwise this would correspond to the vacuous inequality $0 \leqslant 0$, which is not a proper face. We now show that $\tilde{a}$ has exactly one nonzero element. Assume for contradiction that this is not the case, and w.l.o.g. $\tilde{a}_1, \tilde{a}_2 < 0$ (any strictly positive components will not yield a valid inequality for B). This would imply that $(\mathbf{e}^1, h^j) \in B^\star$ for each $j \in [\![d]\!]$ such that $1 \in T^j$, and similarly that $(\mathbf{e}^2, h^j) \in B^\star$ for each $j \in [\![d]\!]$ wherein $2 \in T^j$. However, we could then perform the simple tilting $\tilde{a}_2 \leftarrow 0$ to construct a distinct face with strictly larger support, as now $(\mathbf{e}^2, h^j)$ is supported by the corresponding face for each j such that $2 \in T^j$. Furthermore, as this new constraint does not support $(\mathbf{e}^1, h^j)$ for each j such that $1 \in T^j$, the new face is proper, and thus contradicts the original face F being a facet. Therefore, we can normalize the coefficients to $\tilde{a} = -\mathbf{e}^1$, giving a variable bound constraint on a component of λ which appears in the restriction $\lambda \in \Delta^n$ in (3.2b).

__**Case 2:** $\dim(\tilde{C}) = \dim(C) - 1$__ The fact that $b \perp \tilde{C}$, along with the dimensionality of $\tilde{C}$, implies that $M(\tilde{b}; \mathcal{L}) = \text{span}(\tilde{C})$ is a hyperplane in $\mathcal{L}$. This means we can assume w.l.o.g. that $\tilde{b} = sb^i$ for some $i \in [\![\Gamma]\!]$ and $s \in \{-1, +1\}$. We then compute for each $v \in [\![n]\!]$ that either $a_v = \min_{j:v \in T^j}\{b^i \cdot h^j\}$ if $s = +1$, or $a_v = -\max_{j:v \in T^j}\{b^i \cdot h^j\}$ if $s = -1$.

__**Case 3:** $\dim(\tilde{C}) < \dim(C) - 1$__ We will show that this case cannot occur if F is a general inequality facet. A geometric depiction of the argument is shown in Figure 3-4. Observe that if, w.l.o.g. $\mathbf{e}^1 \notin \text{Proj}_\lambda(\tilde{B})$, then $\tilde{a} \cdot \lambda \leqslant \tilde{b} \cdot z$ is either equal to, or dominated by, the variable bound $\lambda_1 \geqslant 0$. Therefore, we assume that $\text{Proj}_\lambda(\tilde{B}) = \{\mathbf{e}^i\}_{i=1}^n$ for the remainder.

Presume for contradiction that it is indeed the case that F is a facet and $\dim(\tilde{C}) <$



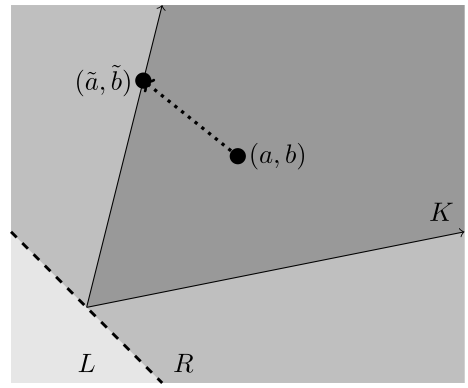


Figure 3-4: **(Left)** As $\dim(L) = 1$, we cannot tilt the inequality (given by coefficients (a,b)) to make one of the inequalities defining K binding, while maintaining feasibility with respect to L and R. **(Right)** This tilting is possible when $\dim(L) > 1$.

$\dim(C) - 1$. As F is a proper face, we know that there is some point in B not supporting F, w.l.o.g. $(\mathbf{e}^1, h^1) \in B\backslash\tilde{B}$. We will take all the remaining extreme points as $B^\star = B\backslash(\tilde{B} \cup \{(\mathbf{e}^1, h^1)\})$.

First, we show that $B^\star \neq \varnothing$. If this where not the case, then $\tilde{B} = B\backslash\{(\mathbf{e}^1, h^1)\}$ necessarily, and this implies that $i = 1$ for each $\{i,j\} \in \Upsilon\backslash\tilde{\Upsilon}$ (recall that $i < j$ notationally). Furthermore, $T^1 \cap T^j \subseteq \{1\}$ for each $j \in [\![2, d]\!]$, else $c^{1,v} \in \tilde{C}$ and $\{1, v\} \in \tilde{\Upsilon}$. Therefore, as Υ is connected by assumption, $\tilde{\Upsilon}$ is "nearly connected" in the sense that $G = \left([\![2,d]\!], \left\{ \{i,j\} \in \tilde{\Upsilon} \;\middle|\; i \neq 1, j \neq 1 \right\}\right)$ is a connected graph. By the same argument as in the beginning of the proof, we conclude that $\operatorname{span}(\tilde{C}) \supseteq \operatorname{aff}(\{h^i\}_{i=2}^d) - h^2$. However, this would imply that $\dim(\tilde{C}) \geqslant \dim(\operatorname{aff}(\{h^i\}_{i=2}^d) - h^2) \geqslant \dim(\operatorname{aff}(\{h^i\}_{i=1}^d) - h^2) - 1 = \dim(\mathcal{L}) - 1 = \dim(C) - 1$, a contradiction of our dimensionality assumption. Therefore, we conclude that $B^\star \neq \varnothing$.

We now define the cone

$$K = \left\{ (a,b) \in \mathbb{R}^n \times \mathcal{L} \;\middle|\; a \cdot \mathbf{e}^v \leqslant b \cdot h^j \;\forall (\mathbf{e}^v, h^j) \in B^\star \right\}$$

and the linear space

$$L = \left\{ (a,b) \in \mathbb{R}^n \times \mathcal{L} \;\middle|\; a \cdot \mathbf{e}^v = b \cdot h^j \;\forall (\mathbf{e}^v, h^j) \in \tilde{B} \right\}.$$

Furthermore, we see that the inequalities defining K cannot be implied equalities. Therefore, as $(\tilde{a}, \tilde{b}) \in K$ and this point strictly satisfies each inequality of K indexed by $B^\star$, we conclude that K is full-dimensional in $\mathbb{R}^n \times \mathcal{L}$, and that $(\tilde{a}, \tilde{b}) \in \mathrm{int}(K)$.

Next, we show that $\dim(L) > 1$. To show this, we start by instead studying $L' = \left\{ b \in \mathcal{L} \;\middle|\; b \cdot c = 0 \, \forall c \in \tilde{C} \right\}$. We can readily observe that $L' = \mathrm{Proj}_b(L)$. Furthermore, as $\mathrm{Proj}_a(\tilde{B}) = \{\mathbf{e}^i\}_{i=1}^n$ from the argument at the beginning of the case, we conclude that the set $\{ a \mid (a, b) \in L \}$ is a singleton. In other words, the values for a are completely determined by the values for b in L. From this, we conclude that $\dim(L) = \dim(L')$. From the definition of L', we see that L' and $\mathrm{span}(\tilde{C})$ form an orthogonal decomposition of $\mathcal{L}$. Therefore, $\dim(\mathcal{L}) = \dim(L') + \dim(\tilde{C})$. Recalling that $\dim(\mathcal{L}) = \dim(C)$, and that we are assuming that $\dim(\tilde{C}) < \dim(C) - 1$, we have that $\dim(L) = \dim(L') = \dim(\mathcal{L}) - \dim(\tilde{C}) = \dim(C) - \dim(\tilde{C}) > 1$, giving the result.

We now show that $K \cap L$ is pointed. To see this, presume for contradiction that there exists a nonzero $(\hat{a}, \hat{b})$ such that $(\hat{a}, \hat{b}), (-\hat{a}, -\hat{b}) \in K \cap L$. However, this would imply that $\hat{a} \cdot \mathbf{e}^v = \hat{b} \cdot h^j$ for all $(\mathbf{e}^v, h^j) \in \tilde{B} \cup B^\star$. Because $B^\star \neq \varnothing$, this implies that $\hat{a} \cdot \lambda \leqslant \hat{b} \cdot z$ is a face strictly containing the facet F, and so must be a non-proper face (i.e. it is additionally supported by $(\mathbf{e}^1, h^1)$ and hence by every point in B). However, this would imply that $\hat{b} \cdot c = 0$ for all $c \in C$, and as $\mathcal{L} = \mathrm{span}(C)$, this would necessitate that $\hat{b} \in \mathcal{L}^\perp$. As $\hat{b} \in \mathcal{L}$ from the definition of K, it follows that $\hat{b} = \mathbf{0}^r$. However, this would imply that $\hat{a} \cdot \lambda = 0$ is valid for B, which cannot be the case unless $\hat{a} = \mathbf{0}^n$, a contradiction. Therefore, $K \cap L$ is pointed.

As $\dim(L) > 1$, we can take some two-dimensional linear subspace $L^2 \subseteq L$ such that $(\tilde{a}, \tilde{b}) \in L^2$. As $(\tilde{a}, \tilde{b}) \in L \cap \mathrm{int}(K)$, it follows that $(\tilde{a}, \tilde{b}) \in L^2 \cap \mathrm{int}(K)$ as well. Similarly, as $K \cap L$ is pointed, it follows that $K^2 = L^2 \cap K$ is pointed as well. Furthermore, as K is full-dimensional in $\mathbb{R}^n \times \mathcal{L}$, K^2 is full-dimensional in $L^2 \subset \mathbb{R}^n \times \mathcal{L}$ (i.e. 2-dimensional).Therefore, a minimal description for it includes the equalities that define L^2, along with exactly two nonempty-face-inducing inequality constraints from the definition of K. Add the single strict inequality $\hat{K}^2 = K^2 \cap \{ (a, b) \in \mathbb{R}^n \times \mathcal{L} \mid a \cdot \mathbf{e}^1 < b \cdot h^1 \}$. As $\tilde{a} \cdot \mathbf{e}^1 < \tilde{b} \cdot h^1$ and $(\tilde{a}, \tilde{b}) \in K^2$, it follows that $\hat{K}^2$ is nonempty and also 2-dimensional, and can be described using only the linear equations defining L^2, the strict inequality $a \cdot \mathbf{e}^1 < b \cdot h^1$, and at least one (and potentially two) of the inequalities previously used to describe K^2. Select one of the defining nonempty-face-inducing inequalities given by $a \cdot \mathbf{e}^v \leqslant b \cdot h^j$, where $(\mathbf{e}^v, h^j) \in B^\star$.

Now construct the restriction $S = \left\{ (a,b) \in \hat{K}^2 \;\middle|\; a \cdot \mathbf{e}^v = b \cdot h^j \right\}$. As $a \cdot \mathbf{e}^v \leqslant b \cdot h^j$ induces a non-empty face on the cone $\hat{K}^2$, S is nonempty. Furthermore, we see that any $(\hat{a}, \hat{b}) \in S$ will correspond to a valid inequality $\hat{a} \cdot \lambda \leqslant \hat{b} \cdot z$ for B with strictly greater support than our original face $\tilde{a} \cdot \lambda \leqslant \tilde{b} \cdot z$. In particular, we see that $(\mathbf{e}^v, h^j) \in B^\star$, i.e. $\tilde{a} \cdot \mathbf{e}^v < \tilde{b} \cdot h^j$, but by construction $\hat{a} \cdot \mathbf{e}^v = \hat{b} \cdot h^j$. Additionally, since $\hat{a} \cdot \mathbf{e}^1 < \hat{b} \cdot h^1$, the corresponding face is proper, which implies that F cannot be a facet, a contradiction. □

We observe three things about Theorem 9. First, it is straightforward to adapt it to the case where $\mathcal{T}$ is not path connected by adding a "dummy" element v to the ground set V and to each set $T \in \mathcal{T}$ (i.e. $T \to T \cup \{v\}$). The family is now path connected, and the resulting formulation can be modified for the original constraint by imposing $\lambda_v \leqslant 0$.

Second, we observe that this geometric characterization is most useful from a practical perspective when computing the set of all hyperplanes spanned by the directions C is easy, and there resulting hyperplanes are not too numerous. This means that we can easy apply the theorem to produce a small, strong MIP formulation for our combinatorial disjunctive constraint.

Third, the description (3.2) may be overly conservative in that some of the constraints (3.2a) may not be facet-inducing, and therefore are not necessary for a valid formulation. This is not the case for the result of Vielma [130], where they are able to prove that for the SOS2 constraint, the inequalities (3.2a) are all facet-inducing. This means the fact that C is spanned by a small number of hyperplanes is a sufficient, but not necessary, condition for the corresponding embedding formulation admitting a small representation.

3.4 Novel MIP formulations for univariate piecewise linear functions

Recall the negative result of Proposition 12, which tells us that we cannot hope produce a MIP formulation for the SOS2 constraint that is smaller or stronger than the ideal, logarithmically-sized `LogIB` formulation of Vielma and Nemhauser. However, we can still hope to apply Theorem 9 to build MIP formulations for univariate piecewise linear functions that are superior to `LogIB` in other ways.

We start by reconstructing `LogIB` a second way, using Theorem 9. We will work with an encoding $H_d^{\mathsf{Log}} \overset{\mathsf{def}}{=} (h^i)_{i=1}^d \subseteq \{0,1\}^{\lceil \log_2(d) \rceil}$ known as Gray codes [120], where adjacent codes differ in exactly one component (i.e. $||h^i - h^{i-1}||_1 = 1$ for all $i \in [\![d-1]\!]$). For the remainder, we will work with a particular Gray code known as the binary reflected Gray code (BRGC), for which we can give a simple recursive definition.

Definition 11. *For $s \in \mathbb{N}$, take the matrices K^s as defined recursively via the sequence $K^1 = (0,1)$ and*

$$K^{s+1} = \begin{pmatrix} K^s & \mathbf{0}^r \\ \mathrm{rev}(K^s) & \mathbf{1}^1 \end{pmatrix},$$

where $\mathrm{rev}(A)$ reverses the rows of the matrix A.

The BRGC with d elements is $H_d^{\mathsf{Log}} \overset{\mathsf{def}}{=} (h^i)_{i=1}^d \subseteq \{0,1\}^r$ for $r = \lceil \log_2(d) \rceil$, where h^i is the i-th row of the matrix K^r.

Vielma used the BRGC, along with his geometric characterization for the SOS2 constraint, to produce a logarithmically-sized ideal formulation for the SOS2 constraint [131], which we will refer to as the *logarithmic embedding* (`Log`) formulation.

Corollary 9 (Corollary 3 [131])**.** *Take the SOS2($d+1$) constraint, given by the family of sets $\mathcal{T}_d^{\mathrm{SOS2}} = (\{i, i+1\})_{i=1}^d$. Take $H_d^{\mathsf{Log}} = (h^i)_{i=1}^d \subseteq \{0,1\}^r$ as the BRGC, where $r = \lceil \log_2(d) \rceil$. For notational convenience, take $h^0 \equiv h^1$ and $h^{d+1} \equiv h^d$. Then the* *following is an ideal formulation for the SOS2 constraint:*

$$\sum_{v=1}^{d+1} \min\{h_i^{j-1}, h_i^j\}\lambda_v \leqslant z_i \leqslant \sum_{v=1}^{d+1} \max\{h_i^{j-1}, h_i^j\}\lambda_v \quad \forall i \in [\![r]\!] \tag{3.3a}$$

$$(\lambda, z) \in \Delta^{d+1} \times \{0,1\}^r. \tag{3.3b}$$

It is constructive to compare this against the `LogIB` formulation given in Chapter 2.6. As first observed by Muldoon [107], this `Log` formulation coincides with the `LogIB` formulation when d is a power-of-two, but this is not necessarily the case otherwise.

Example 5. *Consider the SOS2(4) constraint on $d = 3$ segments. The* `Log` *formulation is*

$$\lambda_3 + \lambda_4 \leqslant z_1, \qquad \lambda_2 + \lambda_3 + \lambda_4 \geqslant z_1 \tag{3.4a}$$

$$\lambda_4 \leqslant z_2, \qquad \lambda_3 + \lambda_4 \geqslant z_2 \tag{3.4b}$$

$$(\lambda, z) \in \Delta^4 \times \{0,1\}^2. \tag{3.4c}$$

In contrast, the `LogIB` *formulation is*

$$\lambda_3 \leqslant z_1, \qquad \lambda_2 + \lambda_3 + \lambda_4 \geqslant z_1 \tag{3.5a}$$

$$\lambda_4 \leqslant z_2, \qquad \lambda_3 + \lambda_4 \geqslant z_2 \tag{3.5b}$$

$$(\lambda, z) \in \Delta^4 \times \{0,1\}^2 \tag{3.5c}$$

Note that we have applied a linear transformation to the formulation using the equation $\sum_{v=1}^4 \lambda_v = 1$ to present the `LogIB` *formulation in a way analogous to (3.4). In this form, we can observe that the first inequality in (3.4a) differs from the first inequality in (3.5a).*

Regardless, both `Log` and `LogIB` are are both ideal and require the same number of variables and constraints.

We will now construct two new encodings by transforming the BRGC. For the remainder of the subsection, assume w.l.o.g. that d is a power-of-two. Otherwise, construct the codes for $\bar{d} = 2^{\lceil \log_2(d) \rceil}$ and take the first d elements of the encoding.

Take $K^r \in \{0,1\}^{d \times r}$ as the matrix whose rows give the BRGC for $d = 2^r$ elements, as in Definition 11. Our first new encoding is produced by transforming K^r to $C^r \in \mathbb{Z}^{d \times r}$, where $C^r_{i,k} = \sum_{j=2}^{i} \left| K^r_{j,k} - K^r_{j-1,k} \right|$ for each $i \in [\![d]\!]$ and $k \in [\![r]\!]$. In words, $C^r_{i,k}$ is the number of times the sequence $(K^r_{1,k}, \ldots, K^r_{i,k})$ changes value, and is monotonic nondecreasing in i. Our second encoding will be $Z^r \in \{0,1\}^{d \times r}$, where $Z^r_k = \mathcal{A}(C^r_k)$ for the linear map $\mathcal{A} : \mathbb{R}^r \to \mathbb{R}^r$ given by $\mathcal{A}(z)_k = z_k - \sum_{\ell=k+1}^{r} z_\ell$ for each component $k \in [\![r]\!]$.

Finally, given the matrices C^r and Z^r, we can define the new encodings as $H^{\mathsf{ZZI}} \stackrel{\text{def}}{=} (h^i)_{i=1}^d$ (resp. $H^{\mathsf{ZZB}} \stackrel{\text{def}}{=} (h^i)_{i=1}^d$), where h^i is the i-th row of C^r (resp. Z^r). We show the encodings for $r = 3$ in Figure 3-5. Furthermore, we can offer a recursive definition of the matrices C^r and Z^r, analogously to Definition 11, and show that both encodings are hole-free and in convex position.

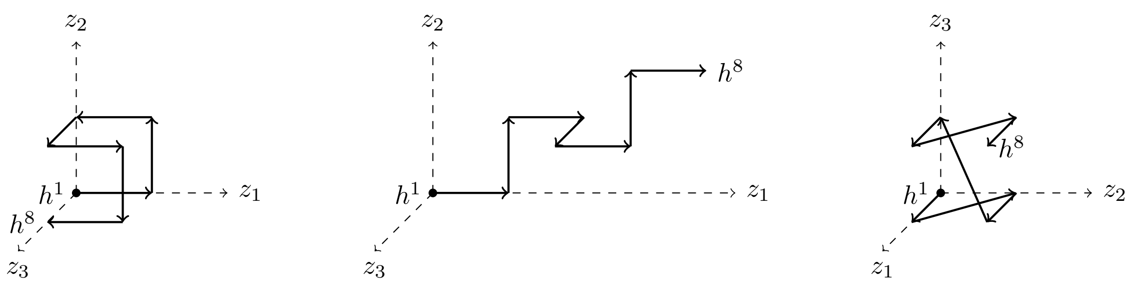


Figure 3-5: Depiction of **(Left)** H_8^{Log}, **(Center)** H_8^{ZZI}, and **(Right)** H_8^{ZZB}. The first row of each is marked with a dot, and the subsequent rows follow along the arrows. The axis orientation is different for H_8^{ZZB} for visual clarity.

Proposition 13. *For each $r \in \mathbb{N}$, $H^{\mathsf{ZZI}}_{2^r}$ and $H^{\mathsf{ZZB}}_{2^r}$ are both hole-free and in convex position. Additionally, $C^1 = Z^1 = (0,1)^T$, and:*

$$C^{r+1} = \begin{pmatrix} C^r & \mathbf{0}^r \\ C^r + \mathbf{1}^r \otimes C^r_r & \mathbf{1}^r \end{pmatrix}$$

$$Z^{r+1} = \begin{pmatrix} Z^r & \mathbf{0}^r \\ Z^r & \mathbf{1}^r \end{pmatrix},$$

where $u \otimes v = uv^T \in \mathbb{R}^{m\times n}$ for any $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$, and $\mathrm{rev}(A)$ *reverses the rows of the matrix A.*

Proof. First, we observe that as $Z^r \in \{0,1\}^{d\times r}$, $H^{\mathsf{ZZB}}_{2^r} \subseteq \{0,1\}^r$ is hole-free and in convex position. Second, we note that $\mathcal{A}$ is an invertible unimodular linear map (i.e. $\mathcal{A}(w) \in \mathbb{Z}^r$ if and only if $w \in \mathbb{Z}^r$), which in turn implies that $H^{\mathsf{ZZI}}_{2^r}$ is hole-free and in convex position. □

Applying Theorem 9 with the new encodings gives two new formulations for SOS2.

Proposition 14. *Take $H^{\mathsf{ZZI}}_d = (h^i)_{i=1}^d$, along with $h^0 \equiv h^1$ and $h^{d+1} \equiv h^d$ for notational simplicity. Let $r = \lceil \log_2(d) \rceil$. Then two ideal formulations for the SOS2$(d+1)$ constraint are*

$$\sum_{v=1}^{d+1} h_i^{v-1}\lambda_v \leqslant z_i \leqslant \sum_{v=1}^{d+1} h_i^v \lambda_v \quad \forall i \in [\![r]\!] \tag{3.6a}$$

$$(\lambda, z) \in \Delta^{d+1} \times \mathbb{Z}^r \tag{3.6b}$$

and

$$\sum_{v=1}^{d+1} h_i^{v-1}\lambda_v \leqslant z_i + \sum_{k=i+1}^{r} 2^{k-i-1} z_k \leqslant \sum_{v=1}^{d+1} h_i^v \lambda_v \quad \forall i \in [\![r]\!] \tag{3.7a}$$

$$(\lambda, z) \in \Delta^{d+1} \times \{0,1\}^r. \tag{3.7b}$$

Proof. Formulations (3.6) and (3.7) correspond to encodings H^{ZZI}_d and H^{ZZB}_d, respectively. The result for (3.6) is direct from Theorem 9, as $C = \{h^{i+1} - h^i\}_{i=1}^{d-1} = \{\mathbf{e}^i\}_{i=1}^r$, where $\mathbf{e}^i$ is the canonical unit vector with support on component i. Therefore, the spanning hyperplanes of C are exactly those given by the normal directions $\{\mathbf{e}^i\}_{i=1}^r$. The result for (3.7) follows by applying the invertible linear map $\mathcal{A}^{-1}$ to the z components of (3.6), and noting that it takes the form $\mathcal{A}^{-1}(z)_i = z_i + \sum_{k=i+1}^r 2^{k-i-1} z_k$ for each $i \in [\![r]\!]$. □

We dub (3.7) the *binary zig-zag* (ZZB) formulation for SOS2, as its associated binary encoding H^{ZZB}_d "zig-zags" through the interior of the unit hypercube (See Figure 3-5). We will refer to formulation (3.6) as the *general integer zig-zag* (`ZZI`) formulation because of its use of general integer encoding $H_d^{\texttt{ZZI}} \subset \mathbb{Z}^r$. We emphasize that `ZZI` and `ZZB` are logarithmically-sized in d and ideal: the same size and strength as the `LogIB` formulation. Additionally, we will see in Chapter 4.1.2 that the new zig-zag formulations enjoy substantially better branching behavior than the existing `Log`/`LogIB` formulations.

3.5 Small MIP formulations for the annulus

We can also return to the annulus constraint of Chapter 1.3.7 to apply Theorem 9 using the three encodings introduced in the previous section. We start by presenting an ideal MIP formulation for the annulus that uses $\log_2(d)$ integer variables and $2\log_2(d)$ general inequality constraints for the case where d is a power-of-two.

Proposition 15. *Fix $d = 2^r$ for some $r \in \mathbb{N}$. Take the BRGC $H_d^{\mathsf{Log}} = (h^i)_{i=1}^d \subseteq \{0,1\}^r$, along with $h^{d+1} \equiv h^1$ for notational simplicity. Then $(\lambda, z) \in Q(\mathcal{P}(\mathcal{T}_d^{\mathrm{ann}}), H_d^{\mathsf{Log}})$ if and only if*

$$\sum_{i=1}^d \min\{h_k^i, h_k^{i+1}\}(\lambda_{2i-1} + \lambda_{2i}) \leqslant z_k \quad \forall k \in [\![r]\!] \tag{3.8a}$$

$$\sum_{i=1}^d \max\{h_k^i, h_k^{i+1}\}(\lambda_{2i-1} + \lambda_{2i}) \geqslant z_k \quad \forall k \in [\![r]\!] \tag{3.8b}$$

$$(\lambda, z) \in \Delta^{2d} \times \mathbb{R}^r. \tag{3.8c}$$

Proof. The result follows from Theorem 9 after observing that $\Upsilon = \{i, i+1\}_{i=1}^{d-1} \cup \{1, d\}$ and therefore that $C = \{\pm \mathbf{e}^k\}_{k=1}^r$, as the binary reflected Gray code is cyclic ($h^d - h^1 = \mathbf{e}^1$). □

Observe that this formulation looks extremely similar to formulation (3.3) for the SOS2 constraint: this is a result of the fortuitous fact that $h^d - h^1 = \mathbf{e}^1$, which holds for the BRGC when d is a power-of-two, but not necessarily for other choices of Gray codes or other values of d. Indeed, it is also not the case when using the general integer zig-zag encoding. However, we are still able to use the general integer zig-zag encoding to produce an ideal MIP formulation with $\log_2(d)$ integer variables and $\mathcal{O}(\log^2(d))$ general inequality constraints.

Proposition 16. *Fix $d = 2^r$ for some $r \in \mathbb{N}$. Select the general integer zig-zag encoding $H_d^{\texttt{ZZI}} = (h^i)_{i=1}^d$, where we use $h^{d+1} \equiv h^1$ for notational convenience. Then $(\lambda, z) \in Q(\mathcal{P}(\mathcal{T}_d^{\text{ann}}), H_d^{\texttt{ZZI}})$ if and only if*

$$\sum_{i=1}^{d} \min\{h_k^i, h_k^{i+1}\}(\lambda_{2i-1} + \lambda_{2i}) \leqslant z_k \quad \forall k \in \llbracket r \rrbracket \tag{3.9a}$$

$$\sum_{i=1}^{d} \max\{h_k^i, h_k^{i+1}\}(\lambda_{2i-1} + \lambda_{2i}) \geqslant z_k \quad \forall k \in \llbracket r \rrbracket \tag{3.9b}$$

$$\sum_{i=1}^{d} \min\left\{\frac{h_k^i}{2^\ell} - \frac{h_\ell^i}{2^k}, \frac{h_k^{i+1}}{2^\ell} - \frac{h_\ell^{i+1}}{2^k}\right\}(\lambda_{2i-1} + \lambda_{2i}) \leqslant \frac{z_k}{2^\ell} - \frac{z_\ell}{2^k} \quad \forall \{k, \ell\} \in [r]^2 \tag{3.9c}$$

$$\sum_{i=1}^{d} \max\left\{\frac{h_k^i}{2^\ell} - \frac{h_\ell^i}{2^k}, \frac{h_k^{i+1}}{2^\ell} - \frac{h_\ell^{i+1}}{2^k}\right\}(\lambda_{2i-1} + \lambda_{2i}) \geqslant \frac{z_k}{2^\ell} - \frac{z_\ell}{2^k} \quad \forall \{k, \ell\} \in [r]^2 \tag{3.9d}$$

$$(\lambda, z) \in \Delta^{2d} \times \mathbb{R}^r. \tag{3.9e}$$

Proof. The result follows from Theorem 9. As $\Upsilon = \{i, i+1\}_{i=1}^{d-1} \cup \{1, d\}$, it follows that $C = \{\mathbf{e}^k\}_{k=1}^r \cup \{w \equiv (2^{r-1}, 2^{r-2}, \ldots, 2^0)\}$. We have that $B = \{\mathbf{e}^k\}_{k=1}^r$ induce all hyperplanes spanned by the vectors $C \backslash \{w\} = \{\mathbf{e}^k\}_{k=1}^r$. Now consider each hyperplane spanned by $\hat{C} = \{\mathbf{e}^k\}_{k \in I} \cup \{w\} \subset C$, where $\hat{I} \subseteq I$. As $|C| = r + 1$ and $\dim(C) = r$, we must have $|I| = r - 2$, i.e. that there are distinct indices $k, \ell \in \llbracket r \rrbracket \backslash I$ where $I \cup \{k, \ell\} = \llbracket r \rrbracket$. We may then compute that the corresponding hyperplane is given by the normal direction $b^{k,\ell} = 2^{-\ell}\mathbf{e}^k - 2^{-k}\mathbf{e}^\ell$. Therefore, we have that the set $B = \{\mathbf{e}^k\}_{k=1}^r \cup \{b^{k,\ell}\}_{\{k,\ell\} \in [r]^2}$ suffices for the conditions of Theorem 9, giving the result. □

We have stated this result only for the general integer zig-zag encoding, and only for the case that d is a power-of-two. However, an analogous result holds if we use either of the BRGC or the binary zig-zag encoding instead, and also if d is not a power-of-two. In particular, we can recover three distinct ideal MIP formulations with $\lceil \log_2(d) \rceil$ integer variables and $\mathcal{O}(\log^2(d))$ general inequality constraints.

We close by investigating the branching behavior of different embedding formulations for the annulus. Consider an instance of the annulus with $d = 8$ quadrilaterals with $\underline{S} = 2$ and $\overline{S} = 3$. Both formulation (3.8) and (3.9) are ideal, and so they project onto the convex hull of the feasible region; see Figure 3-6. However, we can compare the two formulations after branching: that is, after we change variable bounds on some of the integer variables, as is done in a branch-and-bound algorithm.

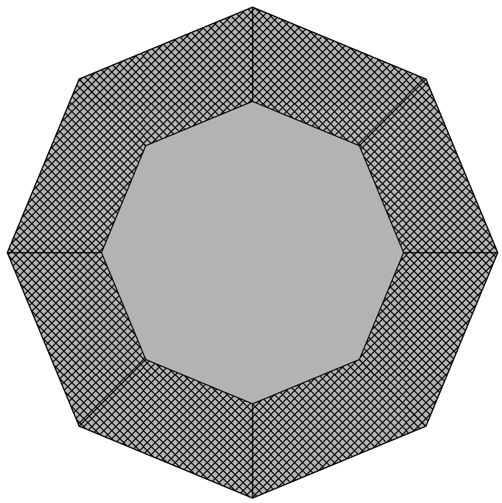


Figure 3-6: Annulus with $d = 8$ quadrilateral pieces (crosshatched), along with LP relaxation of ideal formulation (solid light gray).

For example, in Figure 3-7 we see what happens to formulation (3.8) after branching on z_1: either down ($z_1 \leqslant 0$) or up ($z_1 \geqslant 1$). Both branches produce LP relaxations for the subproblems that are the convex hulls of the quadrilaterals feasible for each (i.e. hereditary sharpness). However, we can see qualitatively that the subproblems after branching have LP relaxations which are not substantially different from the original LP relaxation.

We can contrast this with the behavior of formulation (3.9), as depicted in Figure 3-8, which nearly induces the incremental branching property. Now there are 8 possible branching decisions for z_1, and the corresponding subproblem LP relaxations differ greatly between the different choices. We also observe that the LP relaxations for the subproblems are not always the tightest possible, as sometimes they are strictly larger than the convex hull of the quadrilaterals feasible for the given subproblem.

Figure 3-7: LP relaxation of formulation (3.8) (shaded) after **(Left)** down-branching $z_1 \leqslant 0$, and **(Right)** up-branching $z_1 \geqslant 1$. The quadrilaterals that are feasible for each subproblem are crosshatched.

Finally, if we construct the embedding formulation analogous to (3.9) using the binary zig-zag encoding H_8^{ZZB}, we see in Figure 3-9 that the resulting formulation has extremely pathological branching behavior: both subproblems after branching on z_1 have LP relaxations that remain completely unchanged in x-space! In the following section we will explore these concepts more, as this type of analysis of branching behavior will allow us to explain why the `ZZI` formulation derived earlier computationally outperforms other formulations of comparable size and strength.

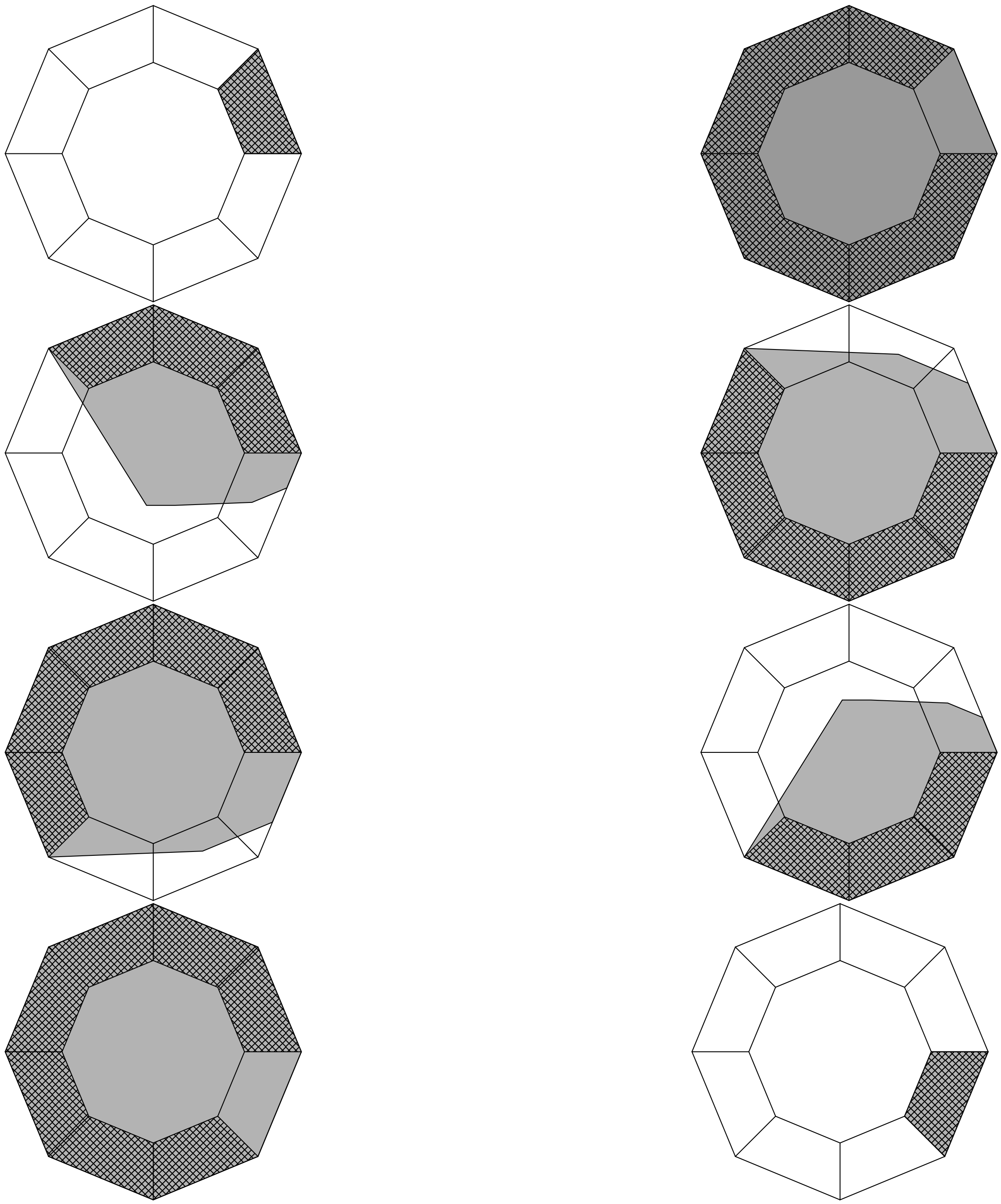


Figure 3-8: LP relaxation of formulation (3.9) (shaded) after **(First row)** down-branching $z_1 \leqslant 0$ or up-branching $z_1 \geqslant 1$; **(Second row)** down-branching $z_1 \leqslant 1$ or up-branching $z_1 \geqslant 2$; **(Third row)** down-branching $z_1 \leqslant 2$ or up-branching $z_1 \geqslant 3$; or **(Last row)** down-branching $z_1 \leqslant 3$ or up-branching $z_1 \geqslant 4$. The quadrilaterals that are feasible for each subproblem are crosshatched.

Figure 3-9: LP relaxation of embedding formulation for the annulus using the binary zig-zag encoding H_8^{ZZB} (shaded) after **(Left)** down-branching $z_1 \leqslant 0$, and **(Right)** up-branching $z_1 \geqslant 1$. The quadrilaterals that are feasible for each subproblem are crosshatched.

Chapter 4

MIP formulations for nonconvex piecewise linear functions.

In this chapter, we focus exclusively on MIP formulations for piecewise linear functions. We will investigate in detail the computational properties of the new MIP formulations we have derived in the previous chapters. We will also present computational modeling tools we have developed that offer a high-level way to formulate piecewise linear functions, in the hopes of making the complex formulations of this thesis more accessible. Afterwards, we turn our attention to high-dimensional piecewise linear functions that arise in deep learning, and present a new strong MIP formulation, along with valid inequalities for more complex structures. In particular, we will see the limits of the combinatorial disjunctive constraint approach in the high-dimensional setting, suggesting the need for the development of new tools and approaches in the future.

4.1 Univariate piecewise linear functions

We start this chapter by presenting extensive computational results comparing the myriad different formulations for the SOS2 constraint and univariate piecewise linear functions. We will see that our new zig-zag (`ZZI` and `ZZB`) formulations presented in Proposition 14 can offer a substantial speed-up over the existing and high-performing

`LogIB` formulation of Vielma and Nemhauser. In the following subsection, we provided a explanation of this phenomena, and a motivation for the exotic choice of encodings used to build the formulations. In particular, we will see how we were able to craft encodings that result in formulations that maintain the size and strength of the `LogIB` formulation, while repairing its degenerate branching behavior.

4.1.1 Univariate computational experiments

To evaluate the new `ZZI` and `ZZB` formulations against existing formulations for univariate piecewise linear functions, we reproduce a variant of the computational experiments of Vielma et al. [133], with the addition of the `ZZB` and `ZZI` formulations. We compare against the `LogIB`, `Log`, `MC`, `CC`, and `DLog` formulations mentioned previously, as well as the *incremental* (`Inc`) formulation which we discuss in more detail in Chapter 4.1.2, and the SOS2 native constraint branching (`SOS2`) implementation.

We evaluate our formulations on single commodity transportation problems of the form

$$
\begin{aligned}
\min_x \quad & \sum_{i \in S} \sum_{j \in D} f_{i,j}(x_{i,j}) \\
\text{s.t.} \quad & \sum_{i \in S} x_{i,j} = d_j \quad \forall j \in D \\
& \sum_{j \in D} x_{i,j} = s_i \quad \forall i \in S \\
& x_{i,j} \geqslant 0 \quad \forall i \in S, j \in D,
\end{aligned}
$$

where we match supply from nodes indexed by S with demand from nodes indexed by D, while minimizing the transportation costs given by the sum of continuous nondecreasing concave univariate piecewise linear functions $f_{i,j}$ for each arc. These instances are publicly available in the `PiecewiseLinearOpt` repository (`https://github.com/joehuchette/PiecewiseLinearOpt.jl`).

We perform a scaling analysis along two axes: the size of the network (i.e. the cardinality of S and D), and the number of segments for each piecewise linear function

$f_{i,j}$. Regarding the first axis, we study both small networks ($|S| = |D| = 10$) and large networks ($|S| = |D| = 20$). Along the second axis, we study families of instances where each piecewise linear function appearing in the objective has $d \in \{6, 13, 28, 59\}$ segments.

Regarding this last choice, we note that although the `Log`/`LogIB` formulation offers great computational performance (particularly for univariate functions with many segments), it has also been observed that logarithmic formulations tend to suffer from a significant performance degradation when the number of segments d is not a power-of-two [35, 107, 108, 135]. Additionally, the `Log` and `LogIB` formulations do not coincide when d is not a power-of-two, as observed in Example 5, offering us a chance to compare the two.

We use CPLEX v12.7.0 with the JuMP algebraic modeling library [48, 92] in the Julia programming language [21] for all computational trials, here and for the remainder of this work. All such trials were performed on an Intel i7-3770 3.40GHz Linux workstation with 32GB of RAM. For each trial, we allow the solver to run for 30 minutes to prove optimality before timing out.

Small networks

We start by studying the small network instances in Table 4.1. For each formulation and each family ($d \in \{6, 13, 28, 59\}$) of 100 instances, we report the average solve time, standard deviation in solve time, and the counts of instances for which the formulation was either the fastest (Win), or was unable to solve to optimality in 30 minutes or less (Fail).

We observe that the `Inc` formulation is superior for smaller instances. Additionally, the `Log` and `LogIB` formulations are roughly equivalent on all families of instances. We observe that the new `ZZI` and `ZZB` formulations are the best performers for larger instances, and one of the two is the fastest formulation for every instance with $d = 59$. Additionally, `ZZI` and `ZZB` both offer a roughly 2x speed-up in mean solve time over `Log` and `LogIB` for most families of instances ($d \in \{13, 28, 59\}$).

We repeat the same experiments with the Gurobi v7.0.2 solver, and report the

d	Metric	`MC`	`CC`	`SOS2`	`Inc`	`DLog`	`Log`	`LogIB`	`ZZB`	`ZZI`
6	Mean (s)	**0.6**	3.8	1.1	**0.6**	1.1	1.4	2.6	1.1	0.9
	Std	**0.3**	4.1	1.5	**0.3**	1.0	1.2	2.4	0.9	0.5
	Win	35	0	7	**46**	5	1	0	4	2
	Fail	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
13	Mean (s)	3.0	71.2	4.5	**1.7**	4.6	4.4	4.2	2.4	2.6
	Std	3.1	152.0	5.8	**0.7**	3.5	3.4	3.0	1.8	1.7
	Win	11	0	9	**47**	11	0	0	15	7
	Fail	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
28	Mean (s)	18.4	178.9	87.4	5.5	11.1	8.8	8.9	5.1	**4.6**
	Std	26.0	359.3	309.3	4.4	8.1	5.6	5.4	3.7	**2.7**
	Win	1	0	6	14	1	0	0	37	**41**
	Fail	**0**	3	3	**0**	**0**	**0**	**0**	**0**	**0**
59	Mean (s)	348.7	541.0	664.3	17.1	19.1	16.3	16.0	9.8	**9.3**
	Std	523.7	610.3	746.4	14.9	11.3	10.3	9.3	6.1	**5.0**
	Win	0	0	0	0	0	0	0	41	**59**
	Fail	7	13	26	**0**	**0**	**0**	**0**	**0**	**0**

Table 4.1: Computational results for univariate transportation problems on small networks with non powers-of-two segments.

results in Table 4.2. Gurobi has a relatively superior implementation of native SOS2 branching that works very well for small and medium instances. However, it performs very poorly on large instances (timing out after 30 minutes on 98 of 100 instances with $d = 59$), and we again observe that the `ZZI` formulation offers a roughly 1.5-2x speedup over the `Log` and `LogIB` formulations for these larger instances. Interestingly, we observe that the `LogIB` formulation also runs 1.5-2x faster than the `Log` formulation on medium and larger instances.

Large networks

In Table 4.3 we present computational results for the large network instances (we repeat the experiments with Gurobi and report the results in Appendix D). Here we observe a roughly 2-3x average speed-up on larger instances for our new formulations over previous methods. Moreover, we highlight that the new formulations have lower variability in solve time, and time out on fewer instances than the existing methods. With $d = 28$, the `SOS2` approach works very well for easier instances, winning on 11 of 100, though its variability is extremely high, timing out on 80 of 100 instances.

d	Metric	MC	CC	SOS2	Inc	DLog	Log	LogIB	ZZB	ZZI
6	Mean (s)	0.8	2.7	**0.2**	0.5	0.7	0.7	0.7	1.0	0.7
	Std	0.4	3.4	**0.2**	**0.2**	0.8	0.7	0.8	0.8	0.6
	Win	0	0	**95**	2	1	1	0	0	1
	Fail	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
13	Mean (s)	4.2	13.4	**0.9**	1.9	4.1	5.2	2.1	2.5	2.7
	Std	4.8	15.3	1.0	**0.9**	4.5	6.0	2.9	2.6	2.3
	Win	0	0	**90**	4	0	0	1	2	3
	Fail	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
28	Mean (s)	30.3	95.2	3.9	6.1	9.2	6.1	**3.3**	4.4	4.4
	Std	43.0	261.3	8.1	5.2	8.7	10.2	**2.7**	4.6	3.7
	Win	0	0	**63**	1	1	7	8	7	13
	Fail	**0**	2	**0**	**0**	**0**	**0**	**0**	**0**	**0**
59	Mean (s)	265.5	372.3	1781.2	24.3	7.3	12.6	9.1	7.5	**6.0**
	Std	409.5	530.0	134.7	23.1	6.7	12.5	9.3	7.2	**5.3**
	Win	0	0	0	0	10	20	16	5	**49**
	Fail	2	8	98	**0**	**0**	**0**	**0**	**0**	**0**

Table 4.2: Computational results with Gurobi for univariate transportation problems on small networks with non powers-of-two segments.

d	Metric	MC	CC	SOS2	Inc	DLog	Log	LogIB	ZZB	ZZI
28	Mean (s)	828.0	1769.3	1498.6	196.9	242.1	332.9	295.8	147.4	**98.0**
	Std	714.3	211.5	646.9	206.8	282.2	430.4	387.9	228.2	**144.4**
	Win	0	0	11	6	1	1	5	10	**66**
	Fail	28	97	80	**0**	1	2	2	1	**0**
59	Mean (s)	1596.9	1800.0	1800.0	793.4	777.1	749.3	753.5	328.7	**273.1**
	Std	475.7	-	-	557.7	593.5	593.3	591.3	383.0	**341.6**
	Win	0	0	0	2	0	1	1	29	**67**
	Fail	82	100	100	11	15	16	17	**2**	**2**

Table 4.3: Computational results for univariate transportation problems on large networks with non powers-of-two segments.

Metric	MC	CC	SOS2	Inc	DLog	Log	LogIB	ZZB	ZZI
Mean (s)	1663.4	1800.0	1800.0	710.6	752.4	793.1	796.0	319.3	**261.4**
Std	**298.7**	-	-	529.9	555.0	570.9	554.4	392.7	316.7
Win	0	0	0	4	0	1	0	27	**53**
Fail	78	85	85	10	15	17	18	2	**1**
Margin	-	-	-	207.0	-	5.6	-	320.1	**348.9**

Table 4.4: Difficult univariate transportation problems on large networks with non powers-of-two segments.

The existing `Inc`, `DLog`, `Log`, and `LogIB` formulations all perform roughly comparably to each other.

In Table 4.4, we focus on those large network problems that are difficult (i.e. no formulation is able to solve the instance in under 100 seconds) but still solvable (i.e. one formulation solves the instance in under 30 minutes). We see that the new formulations are the fastest on 80 of 85 such instances. Moreover, we report the average margin: for those instances for which a given new (resp. existing) formulation is fastest, what is the absolute difference in solve time between it and the fastest existing (resp. new) formulation? In this way, we can measure the absolute improvement offered by our new formulation on an instance-by-instance basis. Here we see that the new formulations offer a substantial improvement on these difficult instances, with an absolute improvement of 5-6 minutes in average solve time over existing methods. Finally, we highlight that there are 5 instances that our new formulations can solve to optimality and for which all existing formulations time out in 30 minutes.

4.1.2 Branching behavior of MIP formulations

As observed by Vielma et al. [133], and corroborated by our computational experiments, the `LogIB` formulation can offer a considerable computational advantage over existing formulations, particularly for univariate piecewise linear functions with many segments (i.e. large d). However, we have seen that the `Log` and `LogIB` formulations are nearly strictly dominated by our new `ZZI` and `ZZB` formulations. Further, all four formulations share the same size (logarithmic in d) and strength (ideal).

To explain the discrepancy, we start with the existing observation that variable branching with `Log` can produce weaker dual bounds than other approaches (e.g. [139]). We will investigate this further with the function given in Example 1.11 with $d = 4$ pieces.

The `Log`/`LogIB` formulation for our univariate piecewise linear function with $d = 4$

pieces is

$$x = \lambda_1 + 2\lambda_2 + 3\lambda_3 + 4\lambda_4 + 5\lambda_5 \tag{4.1a}$$

$$y = 4\lambda_2 + 7\lambda_3 + 9\lambda_4 + 10\lambda_5 \tag{4.1b}$$

$$\lambda_3 \leqslant z_1 \leqslant \lambda_2 + \lambda_3 + \lambda_4 \tag{4.1c}$$

$$\lambda_4 + \lambda_5 \leqslant z_2 \leqslant \lambda_3 + \lambda_4 + \lambda_5 \tag{4.1d}$$

$$(\lambda, z) \in \Delta^5 \times \{0,1\}^2. \tag{4.1e}$$

The corresponding `ZZI` formulation is

$$x = \lambda_1 + 2\lambda_2 + 3\lambda_3 + 4\lambda_4 + 5\lambda_5 \tag{4.2a}$$

$$y = 4\lambda_2 + 7\lambda_3 + 9\lambda_4 + 10\lambda_5 \tag{4.2b}$$

$$\lambda_3 + \lambda_4 + 2\lambda_5 \leqslant z_1 \leqslant \lambda_2 + \lambda_3 + 2\lambda_4 + 2\lambda_5 \tag{4.2c}$$

$$\lambda_4 + \lambda_5 \leqslant z_2 \leqslant \lambda_3 + \lambda_4 + \lambda_5 \tag{4.2d}$$

$$(\lambda, z) \in \Delta^5 \times \mathbb{Z}^2. \tag{4.2e}$$

Finally, the *incremental* (`Inc`) formulation [38, 42, 111] is an ideal formulation for univariate piecewise linear functions whose size scales linearly in d. For the piecewise linear function (1.11), the formulation is

$$x = 1 + \delta_1 + \delta_2 + \delta_3 + \delta_4 \tag{4.3a}$$

$$y = 4\delta_1 + 3\delta_2 + 2\delta_3 + \delta_4 \tag{4.3b}$$

$$\delta_2 \leqslant z_1 \leqslant \delta_1 \tag{4.3c}$$

$$\delta_3 \leqslant z_2 \leqslant \delta_2 \tag{4.3d}$$

$$\delta_4 \leqslant z_3 \leqslant \delta_3 \tag{4.3e}$$

$$\delta_5 \leqslant z_4 \leqslant \delta_4 \tag{4.3f}$$

$$(\delta, z) \in [0,1]^5 \times \{0,1\}^4. \tag{4.3g}$$

We note in passing that we can alternatively derive the `Inc` formulation (modulo an affine transformation of the variables) by the application of Theorem 9 using a "unary" encoding [130, 139]. The `Inc` formulation is designed to induce the traditional SOS2 constraint branching [13].

We will now investigate the relative branching properties of the three formulations. They are all ideal (and therefore sharp as well), and so their LP relaxation projected onto (x, y)-space is $\mathrm{Conv}(\mathrm{gr}(f))$; see Figure 4.1.2. However, we will see that when f is a concave function such as (1.11), `Log`/`LogIB` leads to relaxations after branching that are qualitatively and quantitatively worse than the corresponding relaxations after branching with the `Inc` or `ZZI` formulations.

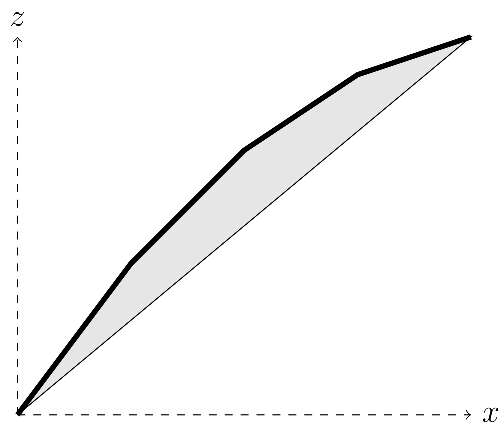


Figure 4-1: The LP relaxation of an ideal formulation for (1.11) (e.g. `Log`/`LogIB`, `Inc`, or `ZZI`) projected onto (x, y)-space.

To quantitatively assess relaxation strength after branching, we consider two metrics. The first is the volume of the projection of the LP relaxation onto (x, y)-space (cf. [88] for a recent work using volume as a metric for formulation quality). The second is the proportion of the domain where the LP relaxation after branching is stronger than the LP relaxation without branching. More formally, Ω is the domain of f in x-space and if F and F' are the projection of the LP relaxations onto (x, y)-space before and after branching, respectively, then we report

$$\frac{1}{\mathrm{Vol}(\Omega)} \mathrm{Vol}\left(\left\{ x \in \Omega \;\middle|\; \min_{(x,y) \in F} z < \min_{(x,y) \in F'} y \right\} \right),$$

which we dub the *strengthened proportion*. We report these statistics for different branching choices for the three formulations in Table 4.5.

Statistic	LP	`Inc` 0 ↓	`Inc` 1 ↑	`Log` 0 ↓	`Log` 1 ↑	`ZZI` 0 ↓	`ZZI` 1 ↑	`ZZI` 1 ↓	`ZZI` 2 ↑
Volume	6	0	2	5.5	0.5	0	3.5	3.5	0
Str. Prop.	0	1	1	0	1	1	0.5	0.5	1

Table 4.5: Metrics for each possible branching decision on z_1 for `Inc`, `Log`, and `ZZI` applied to (1.11). Notationally, $1 \downarrow$ means down-branching on the value 1 (i.e. $z_1 \leqslant 1$, and similarly $2 \uparrow$ implies up-branching on the value 2 (i.e. $z_1 \geqslant 2$.

First, we turn to the `Inc` formulation as depicted in Figure 4-2. The `Inc` formulation also enjoys the hereditary sharpness property. Additionally, the formulation has incremental branching, which has a particularly natural interpretation for univariate piecewise linear functions: after selecting a binary variable z_k for branching (`Inc` has $d-1$ binary variables, so $k \in \llbracket d-1 \rrbracket$), the only points $(x, y) \in \text{gr}(f)$ feasible for the down-branch (resp. up-branch) are those that lie on segments 1 to $k-1$ (resp. k to d). Therefore, after branching down on $z_1 \leqslant 0$, we recover exactly one segment, while when we branch up on $z_1 \geqslant 1$, we recover the convex hull of the remaining pieces, which still shrinks the LP relaxation substantially due to their contiguous ordering.

The combination of ideal strength, hereditary sharpness, and incremental branching mean that the `Inc` formulation tends to perform very well for small and medium-sized instances (as observed in Chapter 4.1.1), until the linear scaling in d of the formulation size starts to dominate in larger instances.

Next, we consider the `Log`/`LogIB` formulation, and refer the reader to Figure 4-3 for an illustration. Down-branching on z_1 (i.e. imposing $z_1 \leqslant 0$) implies that $\lambda_3 = 0$. Up-branching on z_1 (i.e. imposing $z_1 \geqslant 1$) implies that $\lambda_1 = \lambda_5 = 0$. The down branch produces an LP relaxation that is weak and does not substantially change the original LP relaxation before branching. The strengthened proportion is 0, and so when minimizing f, the dual bound will be the same after branching as for the original LP relaxation (assuming both are feasible). Despite this poor branching behavior, the `Log`/`LogIB` formulation is hereditarily sharp. However, we can see in Figure 4-3 that the problem is instead that the feasible segments for the down-branch problem are non-contiguous (and so do not have incremental branching), leading to a large convex relaxation. Therefore, it is reasonable to infer that the high-performance of

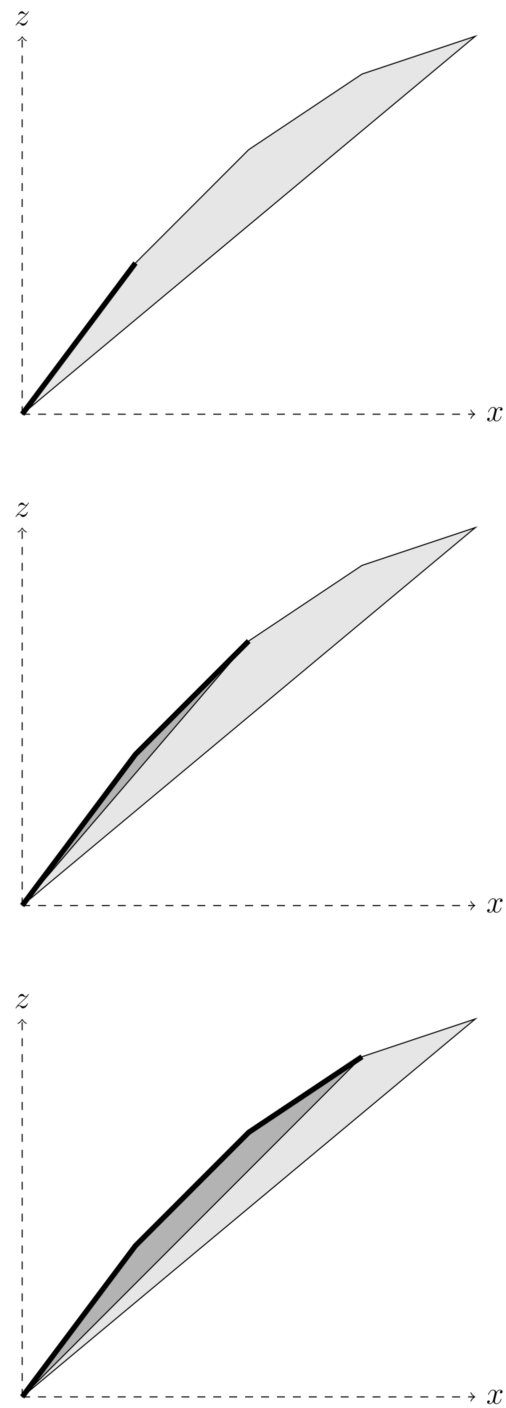


Figure 4-2: The LP relaxation of the `Inc` formulation (4.3) projected onto (x, y)-space, after **(Top Left)** down-branching $z_1 \leqslant 0$, **(Top Right)** up-branching $z_1 \geqslant 1$; **(Center Left)** down-branching $z_2 \leqslant 0$, **(Center Right)** up-branching $z_2 \geqslant 1$; **(Bottom Left)** down-branching $z_3 \leqslant 0$, and **(Bottom Right)** up-branching $z_3 \geqslant 1$.

the `Log` and `LogIB` formulations is due to its strength and small size, in spite of its poor branching behavior.

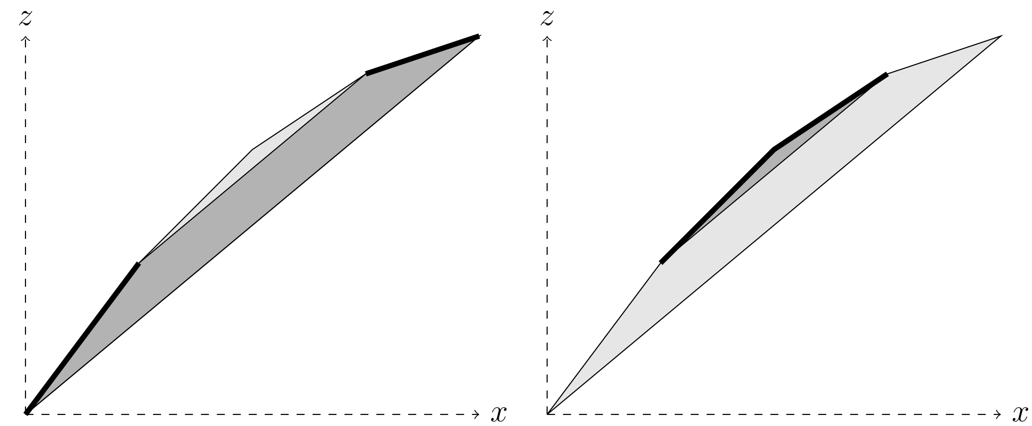


Figure 4-3: The `Log` formulation after **(Left)** down-branching $z_1 \leqslant 0$ and **(Right)** up-branching $z_1 \geqslant 1$.

Finally, we turn to the `ZZI` formulation, which we recall is a general integer MIP formulation. Therefore, we have four possibilities for branching on z_1, depicted in Figure 4-4. Branching $z_1 \leqslant 0$ implies $\lambda_3 = \lambda_4 = \lambda_5 = 0$, while the opposite branch $z_1 \geqslant 1$ implies $\lambda_1 \leqslant \lambda_4 + \lambda_5$. The second branching choice is between $z_1 \leqslant 1$, which implies $\lambda_5 \leqslant \lambda_1 + \lambda_2$, or $z_1 \geqslant 2$, which implies that $\lambda_1 = \lambda_2 = \lambda_3 = 0$. We note that after branching either $z_1 \leqslant 0$ or $z_1 \geqslant 2$, the relaxation is then exact, i.e. the relaxation is equal to exactly one of the segments of the graph of f. Furthermore, when branching either $z_1 \leqslant 1$ or $z_1 \geqslant 1$, we deduce a general inequality on the λ variables that improves the strengthened proportion relative to `Log`. Furthermore, the `ZZI` enjoys the incremental branching property. Therefore, the `ZZI` formulation is both ideal and has incremental branching, but sacrifices hereditary sharpness in lieu of a logarithmic (instead of linear) scaling of its size in d.

As we see qualitatively in Figures 4-3, 4-2, and 4-4, and quantitatively in Table 4.5, the `Inc` formulation offers the best branching behavior of the three ideal formulations, leading to its excellent performance for smaller instances observed in Chapter 4.1.1. In contrast, the `Log`/`LogIB` has very poor branching, but its small size means that it can still perform very well for larger instances, where the `Inc` formulation becomes undesirably large. However, by carefully designing the general integer zig-zag encod-

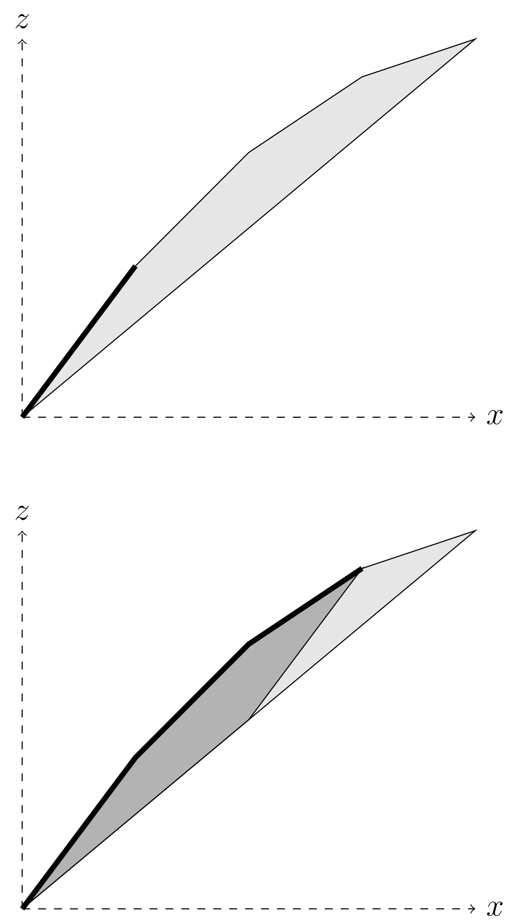


Figure 4-4: The LP relaxation of the `ZZI` formulation (4.2) projected onto (x, y)-space, after **(Top Center)** down-branching $z_1 \leqslant 0$, **(Bottom Center)** up-branching $z_1 \geqslant 1$, **(Top Right)** down-branching $z_1 \leqslant 1$, and **(Bottom Right)** up-branching $z_1 \geqslant 2$.

ing $H_d^{\texttt{ZZI}}$, we are able to maintain the size of the `Log`/`LogIB` formulation, while nearly matching the branching behavior of the `Inc` formulation. In Appendix C, we offer a more complex example with an 8-piece concave piecewise linear function where this effect is even more pronounced.

4.2 New hybrid formulations for bivariate piecewise linear functions

We will now turn our attention to bivariate piecewise linear functions, in the hopes of evaluating the computational performance of the new 6-stencil independent branching formulation of Chapter 2.8.3. Before this, though, we will see how we can blend together embedding and independent branching formulations to create new hybrid formulations that do not sacrifice the strength (i.e. the ideal property) of its comprising parts.

4.2.1 The combination of ideal formulations

Recall that the 6-stencil formulation for bivariate piecewise linear functions is comprised of two stages: the first in terms of two (aggregated) SOS2 constraints, the second via a biclique representation for the "triangle selection" subconstraint. This structure hints at the fact that we could potentially replace the independent branching formulations for the two SOS2 constraints with *any* SOS2 formulation and maintain validity. This means that, for example, we can construct a hybrid formulation for bivariate functions over a grid triangulation by applying the `ZZI` formulation for the aggregated SOS2 constraint along the x_1 and the x_2 dimension, and the 6-stencil independent branching formulation to enforce triangle selection. However, in general the intersection of ideal formulations will not be ideal, with independent branching formulations being a notable exception. Fortunately, the following proposition shows that this preservation of strength is not restricted to independent branching formulations, but holds for any intersection of ideal formulations of combinatorial disjunctive constraints.

Proposition 17. *For each $t \in [\![m]\!]$, take:*

- $U^t = \bigcup_{i=1}^{s_t} P(T^{i,t})$*, where* $\bigcup_{i=1}^{s_i} T^{i,t} = V$.
- $\Pi^t \subseteq \mathbb{R}^V \times \mathbb{R}^{r_t}$ *as an LP relaxation such that* $\{ (\lambda, z^t) \in \Pi^t \mid z^t \in \mathbb{Z}^{r_t} \}$ *is an ideal formulation of* U^t.

Then, an ideal formulation for $\bigcap_{t=1}^m U^t$ *is*

$$\left\{ (\lambda, z^1, \dots, z^m) \;\middle|\; \begin{array}{ll} (\lambda, z^t) \in \Pi^t & \forall t \in [\![m]\!] \\ z^t \in \mathbb{Z}^{r_t} & \forall t \in [\![m]\!] \end{array} \right\}. \tag{4.4}$$

Proof. For simplicity, assume w.l.o.g. that $V = [\![n]\!]$. Let

$$\Pi = \left\{ (\lambda, z^1, \dots, z^m) \in \mathbb{R}^{n+\sum_{i=1}^m r_i} \;\middle|\; (\lambda, z^t) \in \Pi^t \; \forall t \in [\![m]\!] \right\}$$

be the LP relaxation of (4.4). Because the original formulations are ideal (and therefore also sharp), we have

$$\operatorname{Proj}_\lambda(\Pi) = \bigcap_{t=1}^m \operatorname{Proj}_\lambda(\Pi^t) = \bigcap_{t=1}^m \operatorname{Conv}(U^t) \subseteq \Delta^n = \operatorname{Conv}\left(\bigcap_{t=1}^m U^t\right),$$

and hence (4.4) is sharp, as $\operatorname{Proj}_\lambda(\Pi) = \Delta^n$.

To show (4.4) is also ideal, consider any point $(\hat{\lambda}, \hat{z}^1, \dots, \hat{z}^m) \in \Pi$. First, we show that if this point is extreme, then $\hat{\lambda} = \mathbf{e}^v$ for some $v \in [\![n]\!]$. Consider some point where $\hat{\lambda}$ is fractional; w.l.o.g., presume that $0 < \hat{\lambda}_1, \hat{\lambda}_2 < 1$. Define $\lambda^+ \stackrel{\text{def}}{=} \hat{\lambda} + \epsilon \mathbf{e}^1 - \epsilon \mathbf{e}^2$ and $\lambda^- \stackrel{\text{def}}{=} \hat{\lambda} - \epsilon \mathbf{e}^1 + \epsilon \mathbf{e}^2$ for sufficiently small $\epsilon > 0$; clearly $\hat{\lambda} = \frac{1}{2}\lambda^+ + \frac{1}{2}\lambda^-$. We would like to construct points $z^{t,+}$ and $z^{t,-}$ for each $t \in [\![m]\!]$ such that $\hat{z}^t = \frac{1}{2} z^{t,+} + \frac{1}{2} z^{t,-}$, and such that $(\lambda^+, z^{t,+}), (\lambda^-, z^{t,-}) \in \Pi^t$. Then $(\hat{\lambda}, \hat{z}^1, \dots, \hat{z}^m) = \frac{1}{2}(\lambda^+, \hat{z}^{1,+}, \dots, \hat{z}^{m,+}) + \frac{1}{2}(\lambda^-, \hat{z}^{1,-}, \dots, \hat{z}^{m,-})$ is the convex combination of two other feasible points for Π, and so is not extreme.

For a given $t \in [\![m]\!]$, define $E^t = \left\{ (k, h) \;\middle|\; (\mathbf{e}^k, h) \in \operatorname{ext}(\Pi^t) \right\}$, which is equivalent to the set of all extreme points of Π^t. As $(\hat{\lambda}, \hat{z}^t) \in \Pi^t$, there must exist some $\gamma^t \in \Delta^{E^t}$

where $(\hat{\lambda}, \hat{z}^t) = \sum_{(k,z)\in E^t} \gamma^t_{(k,z)}(\mathbf{e}^k, h)$. As $1, 2 \in \text{supp}(\hat{\lambda})$, there must exist some $\tilde{h}^t$ and $\grave{h}^t$ wherein $(1, \tilde{h}^t), (2, \grave{h}^t) \in E^t$ and $0 < \gamma^t_{(1,\tilde{h}^t)}, \gamma^t_{(2,\grave{h}^t)} < 1$. Now define

$$\gamma^{t,\pm}_{(k,h)} = \begin{cases} \gamma^t_{(k,h)} \pm \epsilon & k = 1, h = \tilde{h}^t \\ \gamma^t_{(k,h)} \mp \epsilon & k = 2, h = \grave{h}^t \\ \gamma^t_{(k,h)} & \text{o.w.} \end{cases}$$

Note that, as $\gamma^t \in \Delta^{E^t}$, so is $\gamma^{t,\pm} \in \Delta^{E^t}$. Therefore, we may take

$$\begin{aligned} z^{t,+} &\stackrel{\text{def}}{=} \sum_{(k,h)\in E^t} \gamma^{t,+}_{(k,h)} h = \epsilon\tilde{h}^t - \epsilon\grave{h}^t + \sum_{(k,h)\in E^t} \gamma^t_{(k,h)} h \\ z^{t,-} &\stackrel{\text{def}}{=} \sum_{(k,h)\in E^t} \gamma^{t,-}_{(k,h)} h = -\epsilon\tilde{h}^t + \epsilon\grave{h}^t + \sum_{(k,h)\in E^t} \gamma^t_{(k,h)} h. \end{aligned}$$

Then we may observe that $z^{t,+}, z^{t,-} \in \Pi^t$, and that $\hat{z}^t = \frac{1}{2}z^{t,+} + \frac{1}{2}z^{t,-}$. Now see that

$$\lambda^{\pm} = \sum_{(k,h)\in E^t} \gamma^{t,\pm}_{(k,h)} \mathbf{e}^k = \sum_{(k,h)\in E^t} \gamma^t_{(k,h)} \mathbf{e}^k \pm \epsilon\mathbf{e}^1 \mp \epsilon\mathbf{e}^2 = \hat{\lambda} \pm \epsilon\mathbf{e}^1 \mp \epsilon\mathbf{e}^2$$

Therefore, for each $t \in [\![m]\!]$, we have that $(\lambda^+, z^{t,+}), (\lambda^-, z^{t,-}) \in \Pi^t$, and that $(\hat{\lambda}, \hat{z}^t) = \frac{1}{2}(\lambda^+, z^{t,+}) + \frac{1}{2}(\lambda^-, z^{t,-})$. This implies that $(\lambda^+, h^{1,+}, \ldots, h^{m,+}), (\lambda^+, h^{1,-}, \ldots, h^{m,-}) \in \Pi$ and that $(\hat{\lambda}, \hat{z}^1, \ldots, \hat{z}^m) = \frac{1}{2}(\lambda^+, h^{1,+}, \ldots, h^{m,+}) + \frac{1}{2}(\lambda^-, h^{1,-}, \ldots, h^{m,-})$. Therefore, as our original point is a convex combination of two distinct points also feasible for Π, it cannot be extreme. Therefore, we must have that $\lambda = \mathbf{e}^v$ for some $v \in [\![n]\!]$ for any extreme point of Π.

Now, assume for contradiction that Π has a fractional extreme point. Using property of extreme points just stated, we may assume without loss of generality that this fractional extreme point is of the form $(\mathbf{e}^1, \hat{z}^1, \ldots, \hat{z}^m)$ with $\hat{z}^1 \notin \mathbb{Z}^{r_1}$. As $(\mathbf{e}^1, \hat{z}^1) \in \Pi^1$, then $(\mathbf{e}^1, \hat{z}^1) = \sum_{(v,h)\in E^1} \gamma_{(v,h)}(\mathbf{e}^v, h)$ for some $\gamma \in \Delta^{E^1}$. Also, as Π^1 is ideal and $\hat{z}^1$ is fractional, $(\mathbf{e}^1, \hat{z}^1) \notin \text{ext}(\text{Conv}(\Pi^1))$, and so γ must have at least two non-zero components. But then

$$(\hat{\lambda}, \hat{z}^1, \hat{z}^2, \ldots, \hat{z}^m) = \sum_{(v,h)\in E^1} \gamma_{(v,h)}(\mathbf{e}^1, h, \hat{z}^2, \ldots, \hat{z}^m),$$

a contradiction of the points extremality. Therefore, Π is ideal. □

4.2.2 Bivariate computational experiments

To study the efficacy of the 6-stencil formulation, we perform a computational study on a series of bicommodity transportation problems studied in Section 5.2 of Vielma et al. [133]:

$$\begin{aligned}
\min_{x^1,x^2} \quad & \sum_{i\in S}\sum_{j\in D} f_{i,j}(x^1_{i,j}, x^2_{i,j}) \\
\text{s.t.} \quad & \sum_{i\in S}(x^1_{i,j} + x^2_{i,j}) = d_j \quad \forall j \in D \\
& \sum_{j\in D}(x^1_{i,j} + x^2_{i,j}) = s_i \quad \forall i \in S \\
& x^1_{i,j}, x^2_{i,j} \geqslant 0 \quad \forall i \in S, j \in D,
\end{aligned}$$

The network for each instance is fixed with 5 supply nodes and 5 demand nodes, and the objective functions for these instances are the sum of 25 concave, nondecreasing bivariate piecewise linear functions over grid triangulations with $d_1 = d_2 = \kappa$, where $\kappa \in \{4, 8, 16, 32\}$. The triangulation of each bivariate function is generated randomly, which is the only difference from Vielma et al. [133], where the Union Jack triangulation was used. To handle the generic triangulations, we apply the 6-stencil formulation, coupled with either the `Log`, `ZZB`, or `ZZI` formulation for the SOS2 constraints, taking advantage of Proposition 17 (recall that `Log` and `LogIB` coincide when d is a power-of-two). We compare these new formulations against the `CC`, `MC`, and `DLog` formulations, which readily generalize to bivariate functions. We note in passing that the `Inc` formulation approach also generalizes to bivariate piecewise linear functions, but requires the computation of a Hamiltonian cycle [137], a nontrivial task which may not be practically viable for unstructured triangulations.

κ	Metric	`MC`	`CC`	`DLog`	6-Stencil + `Log`	`ZZB`	`ZZI`
4	Mean (s)	1.4	1.5	0.9	**0.4**	**0.4**	**0.4**
	Std	1.3	1.5	0.6	**0.2**	**0.2**	**0.2**
	Win	0	0	0	29	31	**40**
	Fail	**0**	**0**	**0**	**0**	**0**	**0**
8	Mean (s)	39.3	97.2	12.6	**2.7**	3.0	3.0
	Std	75.0	179.6	9.8	**2.2**	2.4	2.9
	Win	0	0	0	**51**	17	32
	Fail	**0**	**0**	**0**	**0**	**0**	**0**
16	Mean (s)	1370.9	1648.1	352.8	**24.6**	26.5	35.2
	Std	670.4	360.8	499.4	**24.5**	27.4	40.4
	Win	0	0	0	**43**	31	6
	Fail	53	66	6	**0**	**0**	**0**
32	Mean (s)	1800.0	1800.0	1499.6	**133.5**	167.6	246.5
	Std	-	-	475.2	**162.7**	226.7	306.6
	Win	0	0	0	**63**	15	2
	Fail	80	80	50	**0**	**0**	1

Table 4.6: Computational results for bivariate transportation problems on grids of size $= d_1 = d_2$.

In Table 4.6, we see that the new formulations are the fastest on every instance in our test bed. For $\kappa \in \{16, 32\}$, we see an average speed-up of over an order of magnitude over the `DLog` formulation, the best of the existing approaches from the literature. We see that the `Log` 6-stencil formulation wins a plurality or majority of instances for $\kappa \in \{8, 16, 32\}$, and that the hybrid `ZZI` 6-stencil formulation is outperformed by the hybrid `ZZB` 6-stencil formulation by a non-trivial amount on larger instances. Moreover, each of our new formulations is strictly faster than all existing formulations on every instance in the test bed. Finally, we highlight the largest family of instances ($\kappa = 32$), where existing methods are unable to solve 50 of 80 instances in 30 minutes or less, whereas our new formulations can solve all in a matter of minutes, on average.

For completeness, we also perform bivariate computational experiments where κ is not a power-of-two, now adding the `LogIB` 6-stencil formulation as an option for the

κ	Metric	MC	CC	DLog	6-Stencil + Log	LogIB	ZZB	ZZI
6	Mean (s)	9.2	20.8	4.7	1.2	1.5	1.5	**1.1**
	Std	12.0	33.0	3.4	0.7	1.1	1.2	**0.6**
	Win	0	0	0	31	9	12	**48**
	Fail	**0**	**0**	**0**	**0**	**0**	**0**	**0**
13	Mean (s)	1092.9	1507.9	320.3	16.8	**16.5**	17.3	18.1
	Std	729.7	535.4	478.7	18.6	**15.7**	18.6	19.3
	Win	0	0	0	16	**26**	23	15
	Fail	37	58	4	**0**	**0**	**0**	**0**
28	Mean (s)	1768.1	1800.0	1426.2	127.3	131.2	**113.4**	192.7
	Std	139.6	-	513.5	174.5	188.7	**129.7**	254.9
	Win	0	0	0	20	26	**31**	3
	Fail	75	80	46	**0**	**0**	**0**	**0**

Table 4.7: Computational results for transportation problems whose objective function is the sum of bivariate piecewise linear objective functions on grids of size $\kappa = d_1 = d_2$, when κ is not a power-of-two.

SOS2 constraints. We generate the instances by taking each piecewise linear function and randomly dropping $\log_2(\kappa) - 1$ gridpoints from the interior of the domain along each axis. We present the results in Table 4.7. In comparison to the powers-of-two experiment in Table 4.6, we still observe that all new formulations dominate the existing approaches on every instance. However, we now see that the `ZZB` hybrid formulation is the best performing formulation on the largest instances ($\kappa = 28$). There is no significant difference between the `Log` and `LogIB` 6-stencil formulations.

Additionally, we repeat these experiments with the Gurobi solver, which we include in Appendix D. The results are largely the same as with the CPLEX solver, with the 6-stencil formulation winning on every instance in the test bed, and solving over an order of magnitude faster on larger instances than the other approaches.

4.2.3 Optimal independent branching schemes

The 6-stencil formulation is quite small, requiring only $\lceil \log_2(d_1) \rceil + \lceil \log_2(d_2) \rceil + 6$ integer variables. As discussed in Chapter 2.8.1, this is within a constant additive factor of our lower bound of $\lceil \log_2(d_1) + \log_2(d_2) + 1 \rceil$. We also know that the 6-stencil formulation is not always the smallest possible independent branching formulation:

κ	Metric	Log+6S	Log+Opt	ZZB+6S	ZZB+Opt	ZZI+6S	ZZI+Opt
4	Mean (s)	0.420	0.408	0.424	0.406	0.414	0.395
	Sample std	0.217	0.289	0.256	0.295	0.203	0.278
	Win	7	24	17	16	11	25
8	Mean (s)	2.791	2.805	3.021	3.077	3.141	3.178
	Sample std	2.390	2.258	2.400	2.185	3.129	2.492
	Win	26	26	10	12	13	13

Table 4.8: Comparison of bivariate piecewise-linear grid triangulation formulations using the 6-stencil approach (6S) against an optimal triangle selection formulation (Opt) on grids of size $\kappa = d_1 = d_2$.

for the Union Jack triangulation, the construction of Vielma and Nemhauser [133] requires only $\lceil \log_2(d_1) \rceil + \lceil \log_2(d_2) \rceil + 1$ integer variables. This constant term can often have significant impact on the overall computational performance, particularly for problems with many, relatively small piecewise linear functions. For example, if we have a grid triangulation with $d_1 = d_2 = 8$, there is a lower bound (attained by the construction of Vielma and Nemhauser) of 7 levels, while the 6-stencil approach gives a formulation with 12 levels, nearly twice as large. Therefore, it stands to reason that there might be some remaining performance gains to be made by reducing this constant factor.

Fortunately, as we have an equivalency between biclique covers and independent branching formulations (Theorem 3), we can frame the question of finding the smallest formulation as a combinatorial optimization problem. Indeed, we have seen in Proposition 4 that we can solve this using MIP. As an illustration, we construct hybrid formulations that combine a logarithmic formulation for the aggregated SOS2 constraints, along with an optimal biclique representation for the triangle selection subconstraint. In preliminary experiments, we did not observe significant practical advantage for using an optimal representation for the complete triangulation.

In Table 4.8, we report computational experiments for this optimal triangle selection approach on our bivariate test problems with grids of size $d_1 = d_2 = \kappa \in \{4, 8\}$. We study hybrid formulations comprised of one of the `Log`, `ZZB`, and `ZZI` formulations, coupled with either the 6-stencil or an optimal triangle selection formulation. For $\kappa = 4$, we observe that the optimal triangle selection formulations win on 65 of 100 instances, with relatively lower solve times, on average, than their 6-stencil counterparts. For the family of larger instances with $\kappa = 8$, the optimal triangle selection formulations win on 51 of 100 instances. Interestingly, the optimal triangle selection formulations exhibit slightly higher average solve times than the 6-stencil ones, but with a lower variance in solve time.

The MIP formulation for computing optimal triangle selection representations does not scale for instances with $\kappa > 8$, so the evaluation of this approach on larger instances will require new solution techniques for the minimum depth biclique cover problem. However, the subproblems to compute the optimal triangle selection formulations with $\kappa = 4$ solved relatively quickly, on the order of a few seconds. We note that, with $\kappa = 8$, the minimum triangle selection formulation had 3 levels on 54 instances and 4 levels on 46 instances. For the sake of comparison: on those instances where the minimum size representation has 3 levels, the optimal triangle selection formulations had 175 binary variables and 350 general inequality constraints total, while the 6-stencil formulations has 250 binary variables and 500 general inequality constraints.

4.3 Computational tools for piecewise linear modeling: `PiecewiseLinearOpt`

Throughout this thesis, we have investigated a number of possible formulations for optimization problems containing piecewise linear functions. The performance of these formulations can be highly dependent on latent structure of the function and its domain, and there are potentially a number of formulations one may want to try on a given problem instance. However, these formulations can seem quite complex and daunting to a practitioner, especially one unfamiliar with the intricacies and idiosyncrasies of MIP modeling. Anecdotally, we have observed that the complexity of these formulations has driven potential users to simpler but less performant models, or to abandon MIP altogether for other approaches.

This gap between high-performance and accessibility is fundamental throughout

```julia
using JuMP, PiecewiseLinearOpt, CPLEX
model = Model(solver=CplexSolver())
@variable(model, 0 <= x[1:2] <= 4)
xval = [0,1,2,3,4]
fval = [0,4,7,9,10]
z1 = piecewiselinear(model, x, xval, fval, method=:Log)
f(u,v) = 2*(u-1/3)^2 + 3*(v-4/7)^4
dx = dy = linspace(0, 1, 9)
pwl = BivariatePWLFunction(dx, dy, f, pattern=:BestFit)
z2 = piecewiselinear(model, x[1], x[2], pwl, method=:ZZI)
@objective(model, Min, z1 + z2)
```

Figure 4-5: `PiecewiseLinearOpt` code showing how to add univariate and bivariate piecewise linear functions to a JuMP model.

optimization. One essential tool to help close the gap is the modeling language, which allows the user to express an optimization problem in a user-friendly, pseudo-mathematical style, and obviates the need to interact with the underlying optimization solver directly. Because they offer a much more welcoming experience for the modeler, algebraic modeling languages have been widely used for decades, with AMPL [55] and GAMS [117] being two particularly storied and successful commercial examples. JuMP [48] is a recently developed open-source algebraic modeling language in the Julia programming language [21] which offers state-of-the-art performance and advanced functionality, and is readily extensible.

To accompany the advanced formulations presented in this thesis, we have created `PiecewiseLinearOpt`, a Julia package that extends JuMP to offer all the formulation options discussed herein through a simple, high-level modeling interface. The package supports continuous univariate piecewise linear functions, and bivariate piecewise linear functions over grid triangulations.

In Figure 4-5, we see sample code for adding piecewise linear functions to JuMP models. After loading the required packages, we define the `Model` object, and add the `x` variables to it. We add the univariate function (1.11) to our model, specifying it in terms of the breakpoints `xval` of the domain, and the corresponding function values `fval` at these breakpoints. We call the `piecewiselinear` function, while using the `Log` formulation. It returns a JuMP variable `z1` which is constrained to lie in the graph $\mathrm{gr}(f)$ of the function, and can then used anywhere in the model, e.g. in the objective function. After this, we add a bivariate piecewise linear function to our model by approximating a nonlinear function on the box domain $[0,1]^2$. We construct a `BivariatePWLFunction` object to approximate it, choosing the triangulation such that it best approximates the function values at the centerpoint of each subrectangle in the grid. We use the `ZZI` formulation along each axis x_1 and x_2; it will automatically use the 6-stencil triangle selection portion of the formulation, as the triangulation is unstructured.

To showcase the `PiecewiseLinearOpt` package in a more practical setting, we will use it to produce lower bounds for the cutting circle problem, as proposed by Rebennack [116]. The cutting circle problem aims to place I circles in the plane, each with fixed radii R_i, in a rectangle without overlap, while minimizing the total area of the bounding rectangle. Mathematically, we can express this as

$$\begin{aligned}
\min_{x,U} \quad & \log(U_1) + \log(U_2) \\
\text{s.t.} \quad & (x_1^i - x_1^j)^2 + (x_2^i - x_2^j)^2 \geqslant (R_i + R_j)^2 \quad \forall i,j \in [I]^2 \\
& R_i \leqslant x_t^i \leqslant U_t - R_i \quad \forall i \in [\![I]\!], t \in \{1,2\} \\
& 0 \leqslant U_t \leqslant 2\sum_{i=1}^{I} R_i \quad \forall t \in \{1,2\}.
\end{aligned}$$

We can easily express this optimization problem in JuMP and use the `PiecewiseLinearOpt` package to discretize the nonconvexities that appear in the objective and the constraints. In Figure 4-6 we see the entirety of the JuMP code needed to solve the relaxation for the cutting circle problem. In particular, we use a uniform discretization for each nonlinearity with `N=9` segments and determine the piecewise linear functions by interpolating the function values at the breakpoints. Note that the code is written agnostic to the choice of formulation for the univariate piecewise linear functions; we can easily perform a comparison among the many options by changing only one line of code (the definition of `METHOD`).

We believe that this exemplifies the value of `PiecewiseLinearOpt`, and modeling languages more generally: it allows a user to quickly and easily write their problem

```julia
using JuMP, PiecewiseLinearOpt, CPLEX
I = 12; R = rand(I); N = 9
model = Model(solver=CplexSolver())
METHOD = :ZZI
@variable(model, x[i=1:I,1:2] >= R[i])
@variable(model, 0 <= U[1:2] <= 2sum(R))
for i in 1:I
    @constraint(model, x[i,1] <= U[1] - R[i])
    @constraint(model, x[i,2] <= U[2] - R[i])
    for j in (i+1):I
        disc = linspace(-2sum(R)+R[i]+R[j], 2sum(R)-R[i]-R[j], N)
        sqr_x1 = piecewiselinear(model, x[i,1]-x[j,1], disc, t->t^2, method=METHOD)
        sqr_x2 = piecewiselinear(model, x[i,2]-x[j,2], disc, t->t^2, method=METHOD)
        @constraint(model, sqr_x1 + sqr_x2 >= (R[i]+R[j])^2)
    end
end
disc = linspace(2maximum(R), 2sum(R), N)
log_U1 = piecewiselinear(model, U[1], disc, log, method=METHOD)
log_U2 = piecewiselinear(model, U[2], disc, log, method=METHOD)
@objective(model, Min, log_U1 + log_U2)
```

Figure 4-6: JuMP code for the cutting circle problem using `PiecewiseLinearOpt`.

as code, and then iterate as-needed to solve more quickly or to add complexity. For example, we can alter the breakpoint values in the code in Figure 4-6 to modify the model to produce feasible solutions and upper bounds on the optimal solution, or to incorporate the number of other advanced alterations as studied by Rebennack [116].

The `PiecewiseLinearOpt` package supports all the formulations presented in this work, and can handle the construction and formulation of both structured or unstructured grid triangulations. All this complexity is hidden from the user, who can embed piecewise linear functions in their optimization problem in a single line of code with the `piecewiselinear` function. We hope that this simple computational tool will make the advanced formulations available for modeling piecewise linear functions more broadly accessible to researchers and practitioners.

4.4 Preliminary extensions: MIP formulations for neural networks

We now turn our attention to high-dimensional piecewise linear functions. This regime is much more complex than what we have studied thus far: there can easily be exponentially many pieces and extreme points, and so just writing down (much less formulating) a high-dimensional piecewise linear functions is a formidable task. Therefore, we will focus on a particular class of high-dimensional piecewise linear functions that arises from the composition of a number of very simple nonlinearities given by the ReLu activation unit.

4.4.1 Existing formulations

There has been a recent surge of interest in the use of MIP formulations for solving optimization problems containing trained neural networks with ReLu activation units. A standard approach to formulating a single ReLu activation would be to start by formulating the two-dimensional set $\mathsf{MAX}(l,u) \stackrel{\text{def}}{=} \{\ (v,y) \in [l,u] \times \mathbb{R}_{\geqslant 0} \mid y = \max\{0,v\}\ \}$. It is possible to construct an non-extended ideal formulation for this set as

$$
\begin{aligned}
y &\geqslant v && (4.5a)\\
y &\leqslant v - l(1-z) && (4.5b)\\
y &\leqslant uz && (4.5c)\\
(v,y,z) &\in \mathbb{R} \times \mathbb{R}_{\geqslant 0} \times \{0,1\}. && (4.5d)
\end{aligned}
$$

From this, it is simple to construct a formulation for ReLu as

$$
\left\{ (x,v,y,z) \;\middle|\; \begin{array}{c} v = w \cdot x + b \\ (v,y) \in \mathsf{MAX}(l,u) \\ L \leqslant x \leqslant U \\ z \in \{0,1\} \end{array} \right\}, \qquad (4.6)
$$

where we select $l = \min\{\, w \cdot x + b \mid L \leqslant x \leqslant U \,\}$ and $u = \max\{\, w \cdot x + b \mid L \leqslant x \leqslant U \,\}$. This is the approach taken recently by a bevy of authors [5, 31, 122, 126]. Note that it is straightforward to project out the v variable. Unfortunately, this formulation is no longer ideal, or even sharp.

Example 6. *Consider the* `ReLu` *activation given by the data* $L = (-1, -1)$*,* $U = (1, 1)$*,* $w = (1, 1)$*, and* $b = 1$*. Then the LP relaxation for formulation* (4.6) *for* `ReLu` *is*

$$
\begin{aligned}
v &= x_1 + x_2 + 1 \\
y &\leqslant v + (1 - z) \\
y &\leqslant 3z \\
y &\geqslant v \\
-1 \leqslant x_i &\leqslant 1 \quad \forall i \in \{1, 2\} \\
(v, y, z) &\in \mathbb{R} \times \mathbb{R}_{\geqslant 0} \times [0, 1].
\end{aligned}
$$

We may compute that the point $(\hat{x}, \hat{v}, \hat{y}, \hat{z}) = ((1, -1), 1, 3/2, 1/2)$ *is feasible for the LP relaxation; however, we have that* $(\hat{x}, \hat{y}) \equiv ((1, -1), 3/2)$ *is not in*

$$\operatorname{Conv}(\{\, (x, y) \in [L, U] \times \mathbb{R}_{\geqslant 0} \mid y = \max\{0, x_1 + x_2 + 1\} \,\},$$

and so the formulation is not sharp.

Anderson et al. [5] present an ideal extended formulation for `ReLu`, though they observe that the formulation does not appear to offer substantial computational improvements in general. We now present an ideal, non-extended formulation for `ReLu`.

4.4.2 An ideal formulation for a single ReLu

For simplicity, we will assume for the remainder that $w \geqslant 0$ and that $L \leqslant \mathbf{0}^\eta \leqslant U$. Note that both are without loss of generality, provided we are allowed to flip components of x and alter the bias term b. A crucial observation is that this simplification allows us to state that $l \equiv \sum_{i=1}^{\eta} w_i L_i + b$ and $u \equiv \sum_{i=1}^{\eta} w_i U_i + b$.

Our construction works by expressing $\texttt{ReLu}$ as the union of two closely related sets: $\texttt{ReLu} = \mathrm{Proj}_{x,y}(\Gamma^0 \cup \Gamma^1)$, where

$$\Gamma^0 = \left\{ (x, y, z) \;\middle|\; \begin{array}{c} L \leqslant x \leqslant U \\ y = 0 \\ z = 0 \\ w \cdot x + b \leqslant 0 \end{array} \right\}$$
$$\Gamma^1 = \left\{ (x, y, z) \;\middle|\; \begin{array}{c} L \leqslant x \leqslant U \\ y \geqslant 0 \\ z = 1 \\ w \cdot x + b = y \end{array} \right\}.$$

We then are able to prove our result, taking inspiration from Hijazi et al. [62, 63].

Proposition 18. *The following is an ideal formulation for* $\texttt{ReLu}$*:*

$$y \geqslant w \cdot x + b \tag{4.7a}$$
$$y \leqslant \sum_{i \in I} w_i x_i - \sum_{i \in I} w_i L_i (1 - z) + \left(b + \sum_{i \notin I} w_i U_i \right) z \quad \forall I \subseteq [\![\eta]\!] \tag{4.7b}$$
$$y \geqslant \sum_{i \in I} w_i x_i - \sum_{i \in I} w_i U_i (1 - z) + \left(b + \sum_{i \notin I} w_i L_i \right) z \quad \forall I \subseteq [\![\eta]\!] \tag{4.7c}$$
$$(x, y, z) \in [L, U] \times \mathbb{R}_{\geqslant 0} \times [0, 1] \tag{4.7d}$$
$$z \in \{0, 1\}. \tag{4.7e}$$

Proof. Take Γ^* as the feasible set for (4.7a-4.7d) (i.e. the LP relaxation for our formulation). We first show that $\mathrm{Conv}(\Gamma^0 \cup \Gamma^1) \subseteq \Gamma^*$. We can check that $\Gamma^0 \subseteq \Gamma^*$, as (4.7b) reduces to $y \leqslant \sum_{i \in I} w_i (x_i - L_i)$ for each $I \subseteq [\![n]\!]$, whose validity is ensured by the bound $x \geqslant L$. Similarly, we can check that $\Gamma^1 \subseteq \Gamma^*$, as in this case (4.7b) reduces $y \leqslant \sum_{i \in I} w_i x_i + \sum_{i \notin I} w_i U_i + b$, which follows from the bound $x \leqslant U$. An analogous argument holds for the constraints in (4.7c), and so $\mathrm{Conv}(\Gamma^0 \cup \Gamma^1) \subseteq \Gamma^*$ follows.

To show that $\Gamma^* \subseteq \mathrm{Conv}(\Gamma^0 \cup \Gamma^1)$, first we assume w.l.o.g. that $w > 0$ (else we can omit any components with zero components from the formulation). Then we construct the ideal `MC` extended formulation:

$$
\begin{aligned}
x &= x^0 + x^1 \\
y &= y^0 + y^1 \\
z &= z^0 + z^1 \\
\lambda^0 + \lambda^1 &= 1 \\
y^0 = z^0 &= 0 \\
w \cdot x^0 + b\lambda^0 &\leqslant 0 \\
L\lambda^0 \leqslant x^0 &\leqslant U\lambda^0 \\
z^1 &= \lambda^1 \\
w \cdot x^1 + b\lambda^1 &= y^1 \\
L\lambda^1 \leqslant x^1 &\leqslant U\lambda^1 \\
\lambda &\geqslant 0
\end{aligned}
$$

Then we identify $x^0 \equiv x - x^1$, $x^1 \equiv \tilde{x}$, $y \equiv y^1$, $z \equiv z^1$, and $\lambda^0 \equiv 1 - z$ to get

$$w \cdot (x - \tilde{x}) + b(1 - z) \leqslant 0 \tag{4.8a}$$

$$L(1 - z) \leqslant x - \tilde{x} \leqslant U(1 - z) \tag{4.8b}$$

$$w \cdot \tilde{x} + bz = y \tag{4.8c}$$

$$Lz \leqslant \tilde{x} \leqslant Uz \tag{4.8d}$$

$$(y, z) \in \mathbb{R}^+ \times [0, 1], \tag{4.8e}$$

where we say that $\Gamma = \{ (x, \tilde{x}, y, z) \mid (4.8) \}$.

We can apply Fourier-Motzkin to achieve our result. First, we rewrite Γ as

$$
\begin{aligned}
&\tilde{x}_\eta \leqslant x_\eta - L_\eta(1-z) \\
&\tilde{x}_\eta \leqslant U_\eta z \\
&\tilde{x}_\eta \leqslant \frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) \\
&\tilde{x}_\eta \geqslant x_\eta - U_\eta(1-z) \\
&\tilde{x}_\eta \geqslant L_\eta z \\
&\tilde{x}_\eta \geqslant \frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) \\
&\tilde{x}_\eta \geqslant \frac{1}{w_\eta}\left(\sum_{i=1}^{\eta-1} w_i(x_i - \tilde{x}_i) + w_\eta x_\eta + b(1-z)\right) \\
&\tilde{x}_i \leqslant x_i - L_\eta(1-z) && \forall i \in [\![\eta-1]\!] \\
&\tilde{x}_i \leqslant U_\eta z && \forall i \in [\![\eta-1]\!] \\
&\tilde{x}_i \geqslant x_i - U_\eta(1-z) && \forall i \in [\![\eta-1]\!] \\
&\tilde{x}_i \geqslant L_\eta z && \forall i \in [\![\eta-1]\!] \\
&(y, z) \in \mathbb{R}_{\geqslant 0} \times [0,1].
\end{aligned}
$$

and eliminate the $\tilde{x}_\eta$ variable:

$$
\begin{aligned}
x_\eta - U_\eta(1-z) &\leqslant x_\eta - L_\eta(1-z) \\
x_\eta - U_\eta(1-z) &\leqslant U_\eta z \\
x_\eta - U_\eta(1-z) &\leqslant \frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) \\
L_\eta z &\leqslant x_\eta - L_\eta(1-z) \\
L_\eta z &\leqslant U_\eta z \\
L_\eta z &\leqslant \frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) \\
\frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) &\leqslant x_\eta - L_\eta(1-z) \\
\frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) &\leqslant U_\eta z \\
\frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) &\leqslant \frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) \\
\frac{1}{w_\eta}\left(\sum_{i=1}^{\eta-1} w_i(x_i - \tilde{x}_i) + w_\eta x_\eta + b(1-z)\right) &\leqslant x_\eta - L_\eta(1-z) \\
\frac{1}{w_\eta}\left(\sum_{i=1}^{\eta-1} w_i(x_i - \tilde{x}_i) + w_\eta x_\eta + b(1-z)\right) &\leqslant U_\eta z \\
\frac{1}{w_\eta}\left(\sum_{i=1}^{\eta-1} w_i(x_i - \tilde{x}_i) + w_\eta x_\eta + b(1-z)\right) &\leqslant \frac{1}{w_\eta}\left(y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i\right) \\
\tilde{x}_i &\leqslant x_i - L_\eta(1-z) \quad \forall i \in [\![\eta-1]\!] \\
\tilde{x}_i &\leqslant U_\eta z \quad \forall i \in [\![\eta-1]\!] \\
\tilde{x}_i &\geqslant x_i - U_\eta(1-z) \quad \forall i \in [\![\eta-1]\!] \\
\tilde{x}_i &\geqslant L_\eta z \quad \forall i \in [\![\eta-1]\!] \\
(y, z) &\in \mathbb{R}_{\geqslant 0} \times [0,1].
\end{aligned}
$$

Now we explicitly impose the variable bounds $L_\eta \leqslant x_\eta \leqslant U_\eta$, and rewrite this system, dropping redundant constraints, as

$$
\begin{aligned}
w_\eta x_\eta - w_\eta U_\eta(1-z) &\leqslant y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i \\
w_\eta L_\eta z &\leqslant y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i \\
y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i &\leqslant w_\eta x_\eta - L_\eta w_\eta (1-z) \\
y - bz - \sum_{i=1}^{\eta-1} w_i \tilde{x}_i &\leqslant U_\eta w_\eta z \\
\sum_{i=1}^{\eta-1} w_i (x_i - \tilde{x}_i) + b(1-z) &\leqslant -L_\eta w_\eta (1-z) \\
\sum_{i=1}^{\eta-1} w_i (x_i - \tilde{x}_i) + w_\eta x_\eta + b(1-z) &\leqslant U_\eta w_\eta z \\
y &\geqslant \sum_{i=1}^{\eta} w_i x_i + b \\
\tilde{x}_i &\leqslant x_i - L_\eta(1-z) \qquad \forall i \in [\![\eta-1]\!] \\
\tilde{x}_i &\leqslant U_\eta z \qquad \forall i \in [\![\eta-1]\!] \\
\tilde{x}_i &\geqslant x_i - U_\eta(1-z) \qquad \forall i \in [\![\eta-1]\!] \\
\tilde{x}_i &\geqslant L_\eta z \qquad \forall i \in [\![\eta-1]\!] \\
(x, y, z) &\in [L, U] \times \mathbb{R}_{\geqslant 0} \times [0, 1].
\end{aligned}
$$

Repeating this procedure to eliminate the remaining components of $\tilde{x}$ gives the desired result.

□

4.4.3 A separation procedure

As the formulation (4.7) has exponentially many constraints, it behooves us to find a way to separate the constraints dynamically as-needed. To do this, we start by observing if that we construct the valid formulation given by (4.7a,4.7d-4.7e), along with the constraints (4.7b) corresponding to the index sets $I = \varnothing$ and $I = [\![n]\!]$, we recover a formulation which is equivalent to (4.6). As shown in Example 6, this formulation is neither ideal nor sharp, so we would like to construct a procedure to generate constraints from the exponentially-sized families (4.7b) and (4.7c) as-needed to separate some point $(\hat{x}, \hat{y}, \hat{z})$ feasible for the incomplete formulation.

We can separate constraints (4.7b) by solving a problem of the form

$$\alpha = \min_{\tau \in \{0,1\}^\eta} \sum_{i=1}^{\eta} \left(w_i \hat{x}_i - w_i L_i (1 - \hat{z}) \right) \tau_i + w_i U_i \hat{z} (1 - \tau_i)$$

If $\alpha < \hat{y} - b\hat{z}$ with corresponding feasible solution $\hat{\tau}$, then the constraint given by the set $\hat{I} = \{ \, i \in [\![n]\!] \mid \hat{\tau}_i = 1 \, \}$,

$$y \leqslant \sum_{i \in \hat{I}} w_i x_i - \sum_{i \in \hat{I}} w_i L_i (1 - z) + \left(b + \sum_{i \notin \hat{I}} w_i U_i \right) z,$$

separates the point. This problem can be solved greedily: add $i \leftarrow \hat{I}$ if

$$\hat{x}_i < U_i \hat{z} + L_i (1 - \hat{z}) = L_i + (U_i - L_i) \hat{z}.$$

Similarly, we may separate constraints (4.7c) by constructing $\hat{I}$ as the set of all $i \in [\![n]\!]$ such that

$$\hat{x}_i > U_i (1 - \hat{z}) + L_i \hat{z} = U_i - (U_i - L_i) \hat{z}.$$

If

$$\hat{y} < \sum_{i \in \hat{I}} w_i \hat{x}_i - \sum_{i \in \hat{I}} w_i U_i (1 - \hat{z}) + \left(b + \sum_{i \notin \hat{I}} w_i L_i \right) \hat{z},$$

then we have separated a constraint in the family (4.7c), and if this inequality does not hold, then no such inequality separates the given point.

Example 7. *We return to the instance in Example 6, where we saw that the formulation (4.6) was not sharp. To do this, we produced the feasible point* $(\hat{x}, \hat{v}, \hat{y}, \hat{z}) = ((1, -1), 1, 3/2, 1/2)$. *Applying our separation procedure for the family (4.7b), for*

$i = 1$ we have

$$\hat{x}_1 = 1 \nleqslant 0 = L_1 + (U_1 - L_1)\hat{z},$$

and for $i = 2$ we have

$$\hat{x}_2 = -1 < 0 = L_1 + (U_1 - L_1)\hat{z}.$$

Therefore we take $I = \{2\}$, and our separated inequality is

$$y \leqslant w_2 x_2 - w_2 L_2 (1 - z) + (b + w_1 U_1) z = x_2 + z + 1$$

Applying the similar construction to the family (4.7c), we get $I = \{1\}$. However, we can check that the resulting inequality

$$y \geqslant x_1 + z - 1$$

is not violated, and so our procedure does not produce a violated constraint in this family.

4.4.4 Valid inequalities for multiple layers

Although we now have an ideal non-extended formulation for a single ReLu activation unit, it is not hard to see that the composition of such units in a multi-layered network will not, in general, also be ideal. Therefore, we can endeavor to construct stronger formulations for the composition of multiple ReLu units. An explicit description of ideal formulations for multiple layers is beyond the scope of this section, but to close the chapter we present families of valid inequalities that can be used to strengthen our formulations.

Take $F_d \stackrel{\text{def}}{=} \{ (x, y, z) \mid (4.6) \}$ as the big-M formulation for a single ReLu activation unit when the input x is d-dimensional. We construct a big-M MIP formulation for two layers at once as

$$\left\{ \left(x,(y^1,\ldots,y^d),(z^1,\ldots,z^d),\tilde{y},\tilde{z}\right) \;\middle|\; \begin{array}{ll} L \leqslant x \leqslant U & \\ (x,y^i,z^i) \in F_\eta & \forall i \in [\![d]\!] \\ \left((y^1,\ldots,y^d),\tilde{y},\tilde{z}\right) \in F_d & \end{array} \right\}. \tag{4.9}$$

Denote this set as `TwoLayer`.

Proposition 19. *For each $j \in [\![d]\!]$, fix a subset $I^j \subseteq [\![\eta]\!]$. Then valid inequalities for* `TwoLayer` *include:*

$$\tilde{y} \geqslant \sum_{j=1}^{d} \tilde{w}_j \left(\sum_{i \in I^j} w_i^j x_i - \sum_{i \in I^j} w_i^j U_i (1-\tilde{z}) + b^j \tilde{z} + \sum_{i \notin I^j} w_i^j L_i \tilde{z} \right) + \tilde{b}\tilde{z} \tag{4.10a}$$

$$\tilde{y} \leqslant \sum_{j=1}^{d} \tilde{w}_j \left(y^j - \sum_{i \in I^j} w_i^j x_i + \sum_{i \in I^j} w_i^j U_i \tilde{z} - b^j (1-\tilde{z}) - \sum_{i \notin I^j} w_i^j L_i (1-\tilde{z}) \right) + \tilde{b}\tilde{z} \tag{4.10b}$$

Proof. **Validity of** (4.10a)

If $\tilde{z} = 0$, the inequality reduces to

$$\tilde{y} \geqslant \sum_{j=1}^{d} \tilde{w}_j \left(\sum_{i \in I^j} w_i^j (x_i - U_i) \right),$$

which follows as $\tilde{y} = 0$, $\tilde{w}_j, w_i^j \geqslant 0$ for each i and j, and $x \leqslant U$. If $\tilde{z} = 1$, then the inequality reduces to

$$\tilde{y} \geqslant \sum_{j=1}^{d} \tilde{w}_j \left(\sum_{i \in I^j} w_i^j x_i + b^j + \sum_{i \notin I^j} w_i^j L_i \right) + \tilde{b},$$

which follows as

$$y^j \geqslant \sum_{i \in I^j} w_i^j x_i + \sum_{i \notin I^j} w_i^j L_i + b^j,$$

$\tilde{w}_j \geqslant 0$ for each j, and we know that

$$\tilde{y} \geqslant \sum_{j=1}^{d} \tilde{w}_j y^j + \tilde{b}$$

when $\tilde{z} = 1$.

Validity of (4.10b)

If $\tilde{z} = 0$, the inequality reduces to

$$\tilde{y} \leqslant \sum_{j=1}^{d} \tilde{w}_j \left(y^j - \sum_{i \in I^j} w_i^j x_i - \sum_{i \notin I^j} w_i^j L_i - b^j \right),$$

which follows as $\tilde{y} = 0$, $\tilde{w}_j, w_i^j \geqslant 0$ for each i and j, and $y^j \geqslant \sum_{i \in I^j} w_i^j x_i + \sum_{i \notin I^j} w_i^j L_i + b^j$. If $\tilde{z} = 1$, then the inequality reduces to

$$\tilde{y} \leqslant \sum_{j=1}^{d} \tilde{w}_j \left(y^j + \sum_{i \in I^j} w_i^j (U_i - x_i) \right) + \tilde{b},$$

which follows as $x \leqslant U$, $\tilde{w}_j, w_i^j \geqslant 0$ for each i and j, and

$$\tilde{y} \leqslant \sum_{j=1}^{d} \tilde{w}_j y^j + \tilde{b}$$

when $\tilde{z} = 1$. □

Chapter 5

Very small formulations and the MIP-with-holes approach.

We have seen in the previous chapter that MIP formulations can be extremely successful at modeling complex disjunctive constraints. Much of this thesis has focused on finding strong formulations that are as small as possible, as size tends to correlate quite strongly with computational efficacy. However, the existential results of Chapter 3 suggest that so-called "logarithmic MIP formulations" are typically the smallest possible: that is, any MIP formulation requires at least a logarithmic number of integer variables, and typically a comparable number of general inequality constraints as well.

It is worth reiterating that this overhead is not strictly necessary: for example, there exist branch-and-bound methods that work directly on the disjunctions (e.g. [14, 86]). However, much of the the success of MIP formulation approaches can be attributed to the immense advances of MIP solvers over the past decades, as their algorithms are now much more complex than a traditional branch-and-bound approach. In particular, the development of a sophisticated theory on cutting planes plays a crucial role in strengthening bounds and substantially reducing the amount of enumeration required [24, 74]. These techniques can easily combine information from multiple disjunctions and other constraints in the optimization problem to provide tighter relaxations and shorten computation time.

Although it is possible to adopt many of these same algorithmic techniques to the constraint programming realm, this can require significant theoretical developments even for very specific structures [44, 78], as well as substantial engineering effort to implement these ideas in the solver. The goal of this chapter is to suggest a middle ground: a slight modification of the traditional MIP framework that allows us to bypass the logarithmic lower bound on formulation size, while maintaining many of the advanced algorithmic techniques developed in the broad area of mixed-integer programming.

Fortunately, there has recently been growing interest in studying the expressive power and computational properties of generalizations of traditional MIP formulations [9, 27, 64]. In particular, our work builds off of the ideas of Bonami et al. [27] for handling holes in integer sets.

As a simple example, consider the disjunctive constraint $x \in D = \{1, 2, 4, 5\}$. This constraint is nearly equivalent to a standard integrality constraint $x \in [1, 5] \cap \mathbb{Z}$, but with a hole in the domain at 3. A traditional MIP formulation for this might introduce a binary variable z and impose the constraints

$$x \geqslant 1z + 4(1 - z) \tag{5.1a}$$

$$x \leqslant 2z + 5(1 - z) \tag{5.1b}$$

$$(x, z) \in \mathbb{Z} \times \{0, 1\}. \tag{5.1c}$$

Bonami et al. handle these holes directly in the original x space, using *wide split disjunctions*. Recall the standard variable branching approach: given a fractional solution $\hat{x} = 2.5$, round $\hat{x}$ and branches on the valid disjunction $x \leqslant 2 \vee x \geqslant 3$ to separate this point. However, this leaves us no way to separate the hole at $\hat{x} = 3$, which is integer but not feasible for the original constraint. A natural way around this is to impose a wide split disjunction of the form $x \leqslant 2 \vee x \geqslant 4$ to separate $\hat{x}$ from D. This is a straightforward change to the branch-and-bound algorithm in a way that does not require any additional integer variables or constraints.

The crucial observation of Bonami et al. is that wide split disjunctions also readily admit standard cutting plane techniques, such as the intersection cut [4]. By combining the slightly modified branch-and-bound algorithm with cutting planes, Bonami et al. observe a considerable computational speed-up compared to a "full" formulation like (5.1) when optimizing over integer sets with holes.

This chapter extends the wide split idea to the mixed-integer setting with the aim of constructing very small formulations for disjunctive constraints. Once we introduce our *MIP-with-holes* framework, we will be able to leverage the geometric formulation construction technique of Theorem 9 to great effect. In particular, we will be able to offer the following very small formulations.

1. **[Proposition 25]** For any combinatorial disjunctive constraint with d alternatives, we can produce an ideal MIP-with-holes formulation with two integer variables, $\mathcal{O}(d)$ general linear inequality constraints, $\mathcal{O}(|V|)$ variable bounds, and one equation.

2. **[Proposition 26]** For the SOS2 constraint on $n = d + 1$ components, we can produce an ideal MIP-with-holes formulation with two integer variables, four general linear inequality constraints, $\mathcal{O}(n)$ variable bounds, and one equation.

3. **[Proposition 27]** For a relaxation of the annulus as a partition of d quadrilaterals, we can produce an ideal MIP-with-holes formulation with two integer variables, six general linear inequality constraints, $\mathcal{O}(d)$ variable bounds, and one equation.

In other words, our new approach allows us to construct ideal formulations for any combinatorial disjunctive constraint with very few (i.e. constant) integer variables and at most a linear number of general linear inequality constraints. Furthermore, by taking advantage of structure, we can further reduce this to a constant number of general inequality constraints for the SOS2 constraint and for the annulus.

However, a formulation with two integer variables implies a two-dimensional encoding, which cannot be hole-free if $d > 4$. Hence, the resulting formulation is not a traditional MIP formulation. In the following section, we present a way to optimize over such representations in a branch-and-bound setting, using branching schemes customized for a particular encoding. The branching schemes we present will use combinations of variable branching, wide axis-aligned split disjunctions (or just *wide variable branching*) a la Bonami et al. [27], and, in certain degenerate cases, general two-term disjunctions (for which cut generation techniques have also been developed in the literature [4, 12, 26, 79]). As a result, both standard and state-of-the-art cutting plane technology can be deployed to strengthen the relaxations of our formulations.

5.1 Branching schemes and MIP-with-holes formulations

We reiterate that traditional MIP formulations are useful because there exist algorithms–and high-quality implementations of those algorithms–that are able to optimize over these representations efficiently in practice. Roughly, these implementations typically work by applying the branch-and-bound method [86], coupled with the judicious application of cutting planes to strengthen the LP relaxation. In this section, we formalize how our generalized notion of a formulation–the MIP-with-holes framework–fits with this standard approach.

Definition 12. *A* branching scheme *is a procedure that, given*

- *a polyhedron* $Q \subset \mathbb{R}^r$*,*
- *an encoding* $H \subset \mathbb{R}^r$*, and*
- *a point* $\hat{z} \in Q$*,*

either verifies that $\hat{z} \in H$*, or outputs two polyhedra* $Q^1, Q^2 \subset \mathbb{R}^r$ *such that*

- $\hat{z} \notin Q^1$ *and* $\hat{z} \notin Q^2$*,*
- $Q \supseteq Q^1 \cup Q^2$*,*
- $Q \cap H = (Q^1 \cap H) \cup (Q^2 \cap H)$*, and*

- $Q^1 \cap Q^2 = \emptyset$.

The first condition verifies that our branching renders our initial point infeasible. The second guarantees that our feasible region does not expand after branching. The third ensures that we do not lose any feasible points. The fourth tells us that our branching scheme must partition all feasible points between the two subproblems.

We note that branching is described solely in terms of the integer variables z, in contrast to a constraint branching approach [13, 45, 44, 78, 128], which would work directly on the original variables x. However, branching schemes map back to our original variable space in a straightforward way: if R is the LP relaxation for our formulation in (x, z)-space, take $Q = \operatorname{Proj}_z(R)$. Then we can construct Q^1 and Q^2 by adding linear inequalities to Q. These inequalities can be trivially (zero) lifted to inequalities for R with support only on the z variables, giving two polyhedra R^1 and R^2 in the original (x, z)-space.

In the branch-and-bound setting, we will call R^1 and R^2 the LP relaxations for the subproblems, and Q^1 and Q^2 the *code relaxations* for the subproblems. In the case that an encoding H has an associated branching scheme, we will say that the corresponding formulation is a *MIP-with-holes formulation* to emphasize that this is a strict generalization of traditional MIP formulations, and that MIP-with-holes formulations retain many of the computational properties of traditional MIP formulations relevant for branch-and-bound and branch-and-cut methods. Moreover, our standard notions of formulation strength carry over directly.

Definition 13. *Take a MIP-with-holes formulation* $F = \{ (x, z) \in R \mid z \in H \}$ *for* $D \subset \mathbb{R}^n$*, given by an LP relaxation* R *and an encoding* H*. We say the formulation is:*

- sharp *if* $\operatorname{Proj}_x(R) = \operatorname{Conv}(D)$.
- ideal *if* $\operatorname{Proj}_z(\operatorname{ext}(R)) = H$.

Although a branch-and-bound method using variable branching may produce exponentially many subproblems, it enjoys a finite termination guarantee when $Z(R)$

is finite. This is not necessarily the case for our more general setting. For example, take $Q = [0, 1]$ and $H = (0, 1)$. Take the branching scheme that, given a fractional point $0 < \hat{z} < 1$, returns the code relaxations $Q^1 = [0, \hat{z}/2]$ and $Q^2 = [\frac{1}{2}(1 + \hat{z}), 1]$. Then the infinite sequence of points $(\frac{1}{2^k})_{k=1}^{\infty}$ will not allow our branching scheme to finitely terminate.

However, there is a straightforward sufficient condition that we may apply to guarantee finite convergence.

Proposition 20. *Take an encoding H and some branching scheme such that each subproblem code relaxation Q^1 and Q^2 is sharp, in the sense that $Q^1 = \mathrm{Conv}(Q^1 \cap H)$ and $Q^2 = \mathrm{Conv}(Q^2 \cap H)$. Then the branch-and-bound method using this branching scheme is finitely terminating.*

Proof. Consider the sharp polyhedra Q' at some stage in the branch-and-bound algorithm; if the initial polyhedron Q is not sharp, apply the branching scheme once so that all remaining subproblems have this property. Take $H' = Q' \cap H$ as the set of codes feasible with respect to Q', and $\hat{z} \in Q'$ as the point provided to the branching scheme. After applying the branching scheme, we produce two subproblems given by the code relaxations Q^1 and Q^2. Take $H^1 = Q^1 \cap H$ and $H^2 = Q^2 \cap H$. See that as Q' is sharp, $\hat{z} \in Q' = \mathrm{Conv}(H')$, and so $\hat{z}$ can be expressed as a convex combination of the elements in H'. As $\hat{z} \notin Q^1 = \mathrm{Conv}(H^1)$ and $\hat{z} \notin Q^2 = \mathrm{Conv}(H^2)$ necessarily, and $H^1 \subseteq H'$ and $H^2 \subseteq H'$, it follows that $H^1 \subsetneq H'$ and $H^2 \subsetneq H'$. This shows that that each subproblem strictly contracts H', and so after a finite number of iterations of recursively applying the branching scheme, each subproblem will either be infeasible ($H' = \emptyset$), or H' will be a singleton, in which case $Q' = H'$ from sharpness and the branching scheme will verify that $\hat{z} \in H$. $\square$

5.2 Choices of encodings

5.2.1 The reflected binary Gray and zig-zag encodings

In Chapter 3.4 we introduced three possible choices of encodings: the reflected binary Gray (H_d^{Log}), binary zig-zag (H_d^{ZZB}), and general integer zig-zag (H_d^{ZZI}) encodings. In Proposition 13 we showed that these are hole-free and in convex position, and so therefore lead to traditional MIP formulations. These three hole-free encodings give us an opportunity to show how traditional variable branching fits into our branching scheme framework. Take, for example, $H \equiv H_d^{\mathsf{ZZI}}$, and consider some point $\hat{z} \in \operatorname{Conv}(H)$. If $\hat{z} \in \mathbb{Z}^r$, this verifies that $\hat{z} \in H$ from the hole-free property. Otherwise, we can select a component $k \in \llbracket r \rrbracket$ which is fractional, i.e. $\hat{z}_k \notin \mathbb{Z}$. Then the two subproblem code relaxations are created by rounding this component: $Q^1 = \{ z \in Q \mid z_k \leqslant \lfloor \hat{z}_k \rfloor \}$ and $Q^2 = \{ z \in Q \mid z_k \geqslant \lceil \hat{z}_k \rceil \}$.

5.2.2 The moment curve encoding

The η-dimensional moment curve is given by the function $m_\eta(t) = (t, t^2, \dots, t^\eta)$. Given $d (\geqslant \eta)$ ordered points $t_1 < t_2 < \cdots < t_d$ on the real line, the corresponding *cyclic polytope* is $\operatorname{Conv}(\{m_\eta(t_i)\}_{i=1}^d)$, a well-studied object [25, 140]. For our purposes, we are interested in constructing encodings that lie along the two-dimensional moment curve: $H_d^{\mathrm{mc}} \stackrel{\mathsf{def}}{=} (m_2(i))_{i=1}^d$. If $d > 2$, then this choice of encoding is not hole-free; for example, $\frac{1}{2}(m_2(1) + m_2(3)) = (2, 5) \in \operatorname{Conv}(H_d^{\mathrm{mc}}) \backslash H_d^{\mathrm{mc}}$. However, the encoding is in convex position, and it is straightforward to check if $\hat{z} \in H_d^{\mathrm{mc}}$. We also see that a linear inequality description of $\Psi_d(l, u) \stackrel{\mathsf{def}}{=} \operatorname{Conv}(\{ z \in H_d^{\mathrm{mc}} \mid l \leqslant z_1 \leqslant u \})$ is

$$z_2 - k^2 \geqslant (2k+1)(z_1 - k) \quad \forall k \in \llbracket l, u-1 \rrbracket \tag{5.2a}$$

$$(u - l)(z_2 - l^2) \leqslant (u^2 - l^2)(z_1 - l). \tag{5.2b}$$

Our branching scheme for the encoding H_d^{mc} starts with a relaxation of the form $Q = \Psi_d(\ell, u)$ for some $\ell, u \in \mathbb{Z}$. Provided that $\hat{z} \notin H_d^{\mathrm{mc}}$, we create two subproblem code relaxations of the form $Q^1 = \Psi_d(\ell, \lfloor \hat{z}_1 \rfloor)$ and $Q^2 = \Psi_d(\lfloor \hat{z}_1 \rfloor + 1, u)$. See Figure 5-1 for an illustration of the branching.

We emphasize that while this branching scheme uses two-term disjunction branching, in nearly every case a (potentially wide) variable branching disjunction is also valid. For example, the variable branching disjunction $z_1 \leqslant 2 \vee z_1 \geqslant 3$ is valid for the point in the left side of Figure 5-1. Indeed, even if $\hat{z} \in \mathbb{Z}^r$, it will often be the case that a wide variable branching disjunction can be applied. This will only fail in very pathological cases: for example, that depicted in the right side of Figure 5-1. This means that the branching portion of the algorithm can proceed using the branching scheme described above, while the cut generation procedure can also use the valid variable branching split disjunctions as well.

5.2.3 A more exotic encoding

Take some positive integer r, along with $d = 4r$, and consider the encoding $H^{\text{ex}}_d \stackrel{\text{def}}{=} (h^i)_{i=1}^d$ where

$$h^{4k-3} = \left(k - r - 1, \frac{1}{2}(k-1)(k-2r-2)\right) \tag{5.3a}$$

$$h^{4k-2} = \left(r - k + 1, \frac{1}{2}(k-1)(k-2r-2)\right) \tag{5.3b}$$

$$h^{4k-1} = \left(r - k + 1, -\frac{1}{2}k(k-2r-1)\right) \tag{5.3c}$$

$$h^{4k} = \left(k - r, -\frac{1}{2}k(k-2r-1)\right) \tag{5.3d}$$

for each $k \in \llbracket r \rrbracket$. We have depicted H^{ex}_{16} in Figure 5-2. Note that we have drawn in dark lines the differences $h^{i+1} - h^i$ between adjacent codes, to emphasize that these directions are all axis-aligned. Additionally, this encoding is in convex position.

Proposition 21. *For any $r \in \mathbb{N}$, the encoding H^{ex}_{4r} is in convex position.*

Proof. The result for $r = 1$ follows from inspection, so presume that $r > 1$. For each point h^i, we propose an inequality $c^i \cdot z \leqslant b^i$ that strictly separates h^i from the

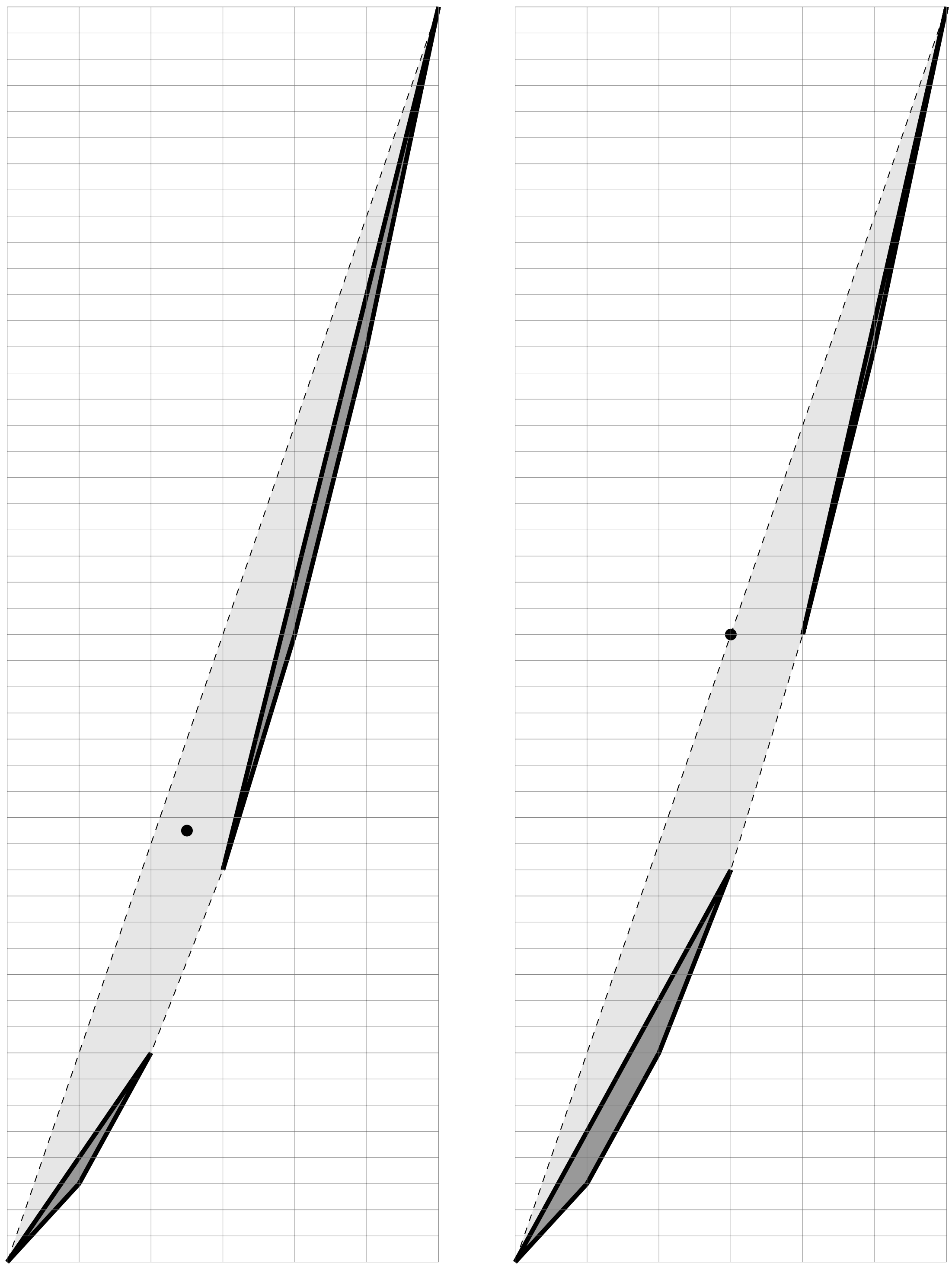


Figure 5-1: Illustration of the branching scheme for the moment curve encoding H_7^{mc}. The original code relaxation in the z-space is shown in the dashed region, and those for the two subproblems are shown in the darker shaded regions. The optimal solution for the original LP relaxation is depicted with a solid dot. We show the branching with a solution $\hat{z}$ that is fractional **(Left)**, and one where $\hat{z} \in \mathrm{Conv}(H_7^{\mathrm{mc}}) \backslash H_7^{\mathrm{mc}}$ and yet there is no valid variable branching disjunction to separate the point **(Right)**.

remaining codes in H_d^{ex}. For each $k \in [\![r]\!]$, the coefficients are

$$
\begin{aligned}
c^{4k-3} &= (-(r-k+2)-(r-k+1), -2) \\
c^{4k-2} &= ((r-k+2)+(r-k+1), -2) \\
c^{4k-1} &= ((r-k+1)+(r-k), 2) \\
c^{4k} &= (-(r-k+1)-(r-k), 2)\,,
\end{aligned}
$$

where $b^i = c^i \cdot h^{i+4}$ for $i \in [\![4]\!]$ and $b^i = c^i \cdot h^{i-4}$ for $i \in [\![5, 4r]\!]$. $\square$

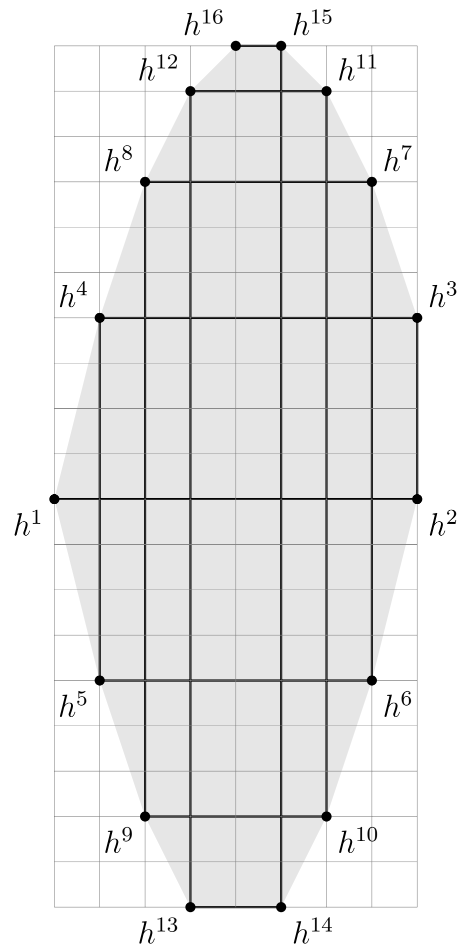


Figure 5-2: The exotic two-dimensional encoding H_{16}^{ex}.

The structure of this encoding also suggests a relatively simple branching scheme. Given a point $\hat{z} \notin H_d^{\text{ex}}$, we consider three cases, depicted in Figure 5-3. In the first case, $\hat{z}_1 \notin \mathbb{Z}$, and we perform standard variable branching: $Q^1 = \{\, z \in Q \mid z_1 \leqslant \lfloor \hat{z}_1 \rfloor \,\}$

and $Q^2 = \{ z \in Q \mid z_1 \geqslant \lceil \hat{z}_1 \rceil \}$. If $\hat{z}_1 \in \mathbb{Z}$, then we consider two other cases. Take $Y = \{ h_2 \mid h \in H_d^{\text{ex}} \}$ as the set of all values the encoding takes in the second component, $\underline{b} = \max \{ t \in Y \mid t < \hat{z}_2 \}$, and $\overline{b} = \min \{ t \in Y \mid t > \hat{z}_2 \}$. Note that $Y \subset \mathbb{Z}$. If $\hat{z}_2 \notin Y$, then we apply a wide variable branching of the form $Q^1 = \{ z \in Q \mid z_2 \leqslant \underline{b} \}$, and $Q^2 = \left\{ z \in Q \mid z_2 \geqslant \overline{b} \right\}$.

The final case remains where $\hat{z}_1 \in \mathbb{Z}$, $\hat{z}_2 \in Y \subset \mathbb{Z}$, and yet $\hat{z} \notin H_d^{\text{ex}}$. In this case, we will branch on a two-term non-parallel disjunction. Take $W(b) = \{ h \in H_d^{\text{ex}} \mid h_2 = b \}$. We take the nearest point to the northeast of $\hat{z}$ as $h^{NE} \in W(\overline{b})$ such that $h_1^{NE} \geqslant \max_{h \in W(\overline{b})} h_1$. Next, take the nearest point to the southwest $h^{SW} \in W(\underline{b})$ such that $h^{SW} \leqslant \max_{h \in W(\underline{b})} h_1$. Take the points directly to the west and east of $\hat{z}$, $h^W, h^E \in W(\hat{z}_2)$ (i.e. $h_1^W < h_1^E$), and we can express the two-term non-parallel disjunction branching with two subproblem code relaxations as $Q^1 = \left\{ z \in Q \mid (h_1^{NE} - h_1^W)(z_2 - h_2^W) \geqslant (h_2^{NE} - h_2^W)(z_1 - h_1^W) \right\}$ and $Q^2 = \left\{ z \in Q \mid (h_1^{SW} - h_1^E)(z_2 - h_2^E) \geqslant (h_2^{SW} - h_2^E)(z_1 - h_1^E) \right\}$.

We emphasize again that this general two-term disjunction branching only needs to be deployed in pathological cases: namely, $\hat{z} \in \mathbb{Z}^2$, with further restrictions on the value for $\hat{z}$. In all other cases, there exist valid split disjunctions (standard or wide) which can be used to separate $\hat{z}$, and for cut generation.

5.3 Very small MIP-with-holes formulations

We are now in a position to use our MIP-with-holes framework to produce very small formulations for disjunctive constraints. We will make extensive use of the following slight modification of Proposition 9, which allows us to construct MIP-with-holes formulations using our geometric construction methods from Chapter 3.

Corollary 10. *Take a family of sets $\mathcal{P} = (P^i)_{i=1}^d$, the disjunctive set $D = \bigcup_{i=1}^d P^i$, and an encoding $H = (h^i)_{i=1}^d$ that is in convex position with an accompanying branching scheme. Then:*

- $D = \bigcup_{i=1}^d \operatorname{Slice}(\operatorname{Em}(\mathcal{P}, H); h^i)$.

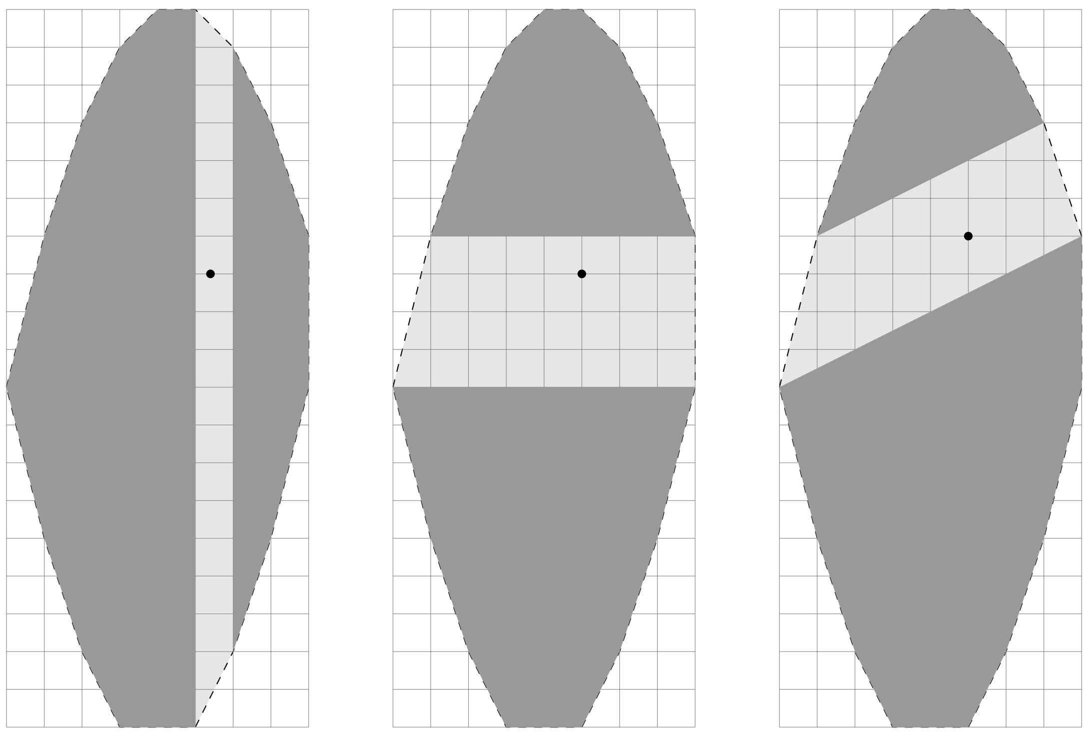


Figure 5-3: Branching scheme for the exotic encoding H_{16}^{mc} when the LP optimal solution for the integer variables $\hat{z}$ has: **(Left)** $\hat{z}_1$ fractional, **(Center)** $\hat{z}_1 \in \mathbb{Z}$ but $\hat{z}_2 \notin Y = \{ h_2 \mid h \in H_{16}^{\mathrm{mc}} \}$, and **(Right)** $\hat{z}_1 \in \mathbb{Z}$, $\hat{z}_2 \in Y$, and yet $\hat{z} \notin H_{16}^{\mathrm{mc}}$. The relaxations for the two subproblems in each are the two shaded regions in each picture.

- $\{ (x, w, z) \in Q(\mathcal{P}, H) \mid z \in H \}$ *is a valid MIP-with-holes formulation for* D.

5.3.1 A big-M MIP-with-holes formulation for any disjunctive set

We start by offering a simple big-M MIP-with-holes formulation that works for any disjunctive constraint. This formulation will not, in general, be ideal or sharp; however, it may be the case that it is simpler to construct, or substantially smaller, than an ideal formulation built through the combinatorial disjunctive constraint approach. It uses only two integer variables and a modest number of constraints.

Proposition 22. *Take* $\mathcal{P} = (P^i = \{ x \in \mathbb{R}^n \mid A^i x \leqslant b^i \})_{i=1}^d$ *as a family of bounded polyhedra, where* $A^i \in \mathbb{R}^{m_i \times n}$ *and* $b^i \in \mathbb{R}^{m_i}$. *Fix constants* $M^i \in \mathbb{R}^{m_i}$ *for each* $i \in [\![d]\!]$ *such that* $M^i_s \geqslant \max_{x \in \bigcup_{k \neq i} P^k} A^k_s x$ *for each* $s \in [\![m_i]\!]$. *Then* $(x, z) \in \operatorname{Em}(\mathcal{P}, H^{\mathrm{mc}}_d)$ *if and only if*

$$A^i x \leqslant b^i + (M^i - b^i)\left(i^2 - 2iz_1 + z_2\right) \quad \forall i \in [\![d]\!] \tag{5.4a}$$

$$z \in H^{\mathrm{mc}}_d. \tag{5.4b}$$

Additionally, we can construct a LP relaxation for the corresponding MIP-with-holes formulation of $\bigcup_{i=1}^d P^i$ *by replacing* (5.4b) *with the constraint* $z \in \Psi_d(1, d)$.

Proof. Consider the constraints (5.4a), given $z = (i, i^2) \in H^{\mathrm{mc}}_d$. The j-th set of constraints in (5.4a) simplifies to

$$A^j x \leqslant \begin{cases} b^j + (M^j - b^j)(i^2 - 2i^2 + i^2) = b^i & j = i \\ b^j + (M^j - b^j)(j^2 - 2j \cdot i + i^2) \equiv \alpha^j & \text{o.w.} \end{cases}$$

As $j^2 - 2i \cdot j + i^2 = (i - j)^2 \geqslant 1$ for each $i, j \in \mathbb{Z}$ with $i \neq j$, we have that $\alpha \geqslant M^j$. Therefore, given $z = (i, i^2) \in H^{\mathrm{mc}}_d$, x satisfies these constraints if and only if $x \in P^i$. $\square$

We compare this formulation against a traditional big-M MIP formulation [130].

Both require $\sum_{i=1}^{d} m_i$ general inequality constraints, along with $\mathcal{O}(d)$ additional constraints to describe either $\Psi_d(1, d)$ or variable bounds on binary variables. However, formulation (5.4) requires only two integer variables, compared to the $\lceil \log_2(d) \rceil$ binary integer variables needed for a traditional big-M MIP formulation.

5.3.2 A very small formulation for the SOS1 constraint

We can also present a simple ideal MIP-with-holes formulation for the SOS1 constraint. Using the moment curve encoding, we may construct a very small formulation with only two integer variables and no general inequality constraints.

Proposition 23. *Take $\mathcal{T}_d^{\text{SOS1}} = (\{i\})_{i=1}^d$ as the SOS1 constraint, and the two-dimensional moment curve encoding H_d^{mc}. Then $Q(\mathcal{P}(\mathcal{T}_d^{\text{SOS1}}), H_d^{\text{mc}})$ is*

$$\left\{ \left(\lambda, \sum_{i=1}^{d} i\lambda_i, \sum_{i=1}^{d} i^2 \lambda_i \right) \middle| \lambda \in \Delta^d \right\}. \tag{5.5}$$

Proof. Immediate, as $\text{Em}(\mathcal{P}(\mathcal{T}), H_d^{MC}) = \{(\mathbf{e}^i, i, i^2)\}_{i=1}^d$. $\square$

We see a stark contrast here with the negative results of Proposition 12, which imply that any MIP formulation for the SOS1 constraint must have at least $\lceil \log_2(n) \rceil$ integer variables. Additionally, we notice that this MIP-with-holes formulation also enjoys favorable incremental branching properties akin to traditional SOS1 branching [13]. Given d distinct points $\{v^i\}_{i=1}^d \subset \mathbb{R}$, consider the simple optimization problem proposed by Yıldız and Vielma:

$$\min_{x,t,\lambda} \quad t \tag{5.6a}$$

$$\text{s.t.} \quad t \geqslant x \tag{5.6b}$$

$$t \geqslant -x \tag{5.6c}$$

$$x = \sum_{i=1}^{d} v^i \lambda_i \tag{5.6d}$$

$$\lambda \in \text{CDC}(\mathcal{T}_d^{\text{SOS1}}). \tag{5.6e}$$

The authors show [139, Proposition 2.1] that an incremental, logarithmic, and unary formulation for (5.6e) can be solved by branching at most once, $\lceil \log_2(d) \rceil$, or $\lceil d/2 \rceil$ times, respectively. It is not hard to see that the MIP-with-holes formulation (5.5), coupled with the moment curve branching scheme, also requires at most one branch to solve to optimality.

5.3.3 Very small formulations for general combinatorial disjunctive constraints

We now return to geometric construction of Theorem 9, which allows us to state a general result: given any combinatorial disjunctive constraint and any two-dimensional encoding in convex position, we can provide an explicit description for a very small ideal MIP-with holes formulation.

Proposition 24. *Take $\mathcal{T} = (T^i \subseteq V)_{i=1}^d$ and let $H = (h^i)_{i=1}^d \subset \mathbb{R}^2$ be any two-dimensional encoding in convex position. Take $b^{i,j} = (c_2^{i,j}, -c_1^{i,j})$ for each $\{i,j\} \in [d]^2$. Then $(\lambda, z) \in Q(\mathcal{P}(\mathcal{T}), H)$ if and only if*

$$\sum_{v \in V} \min_{s : v \in T^s} \{b^{i,j} \cdot h^s\} \lambda_v \leqslant b^{i,j} \cdot z \leqslant \sum_{v \in V} \max_{s : v \in T^s} \{b^{i,j} \cdot h^s\} \lambda_v \quad \forall \{i,j\} \in [d]^2 \tag{5.7a}$$

$$(\lambda, z) \in \Delta^V \times \mathbb{R}^2. \tag{5.7b}$$

Proof. The result follows from Theorem 9. Assume for simplicity that $V = [\![n]\!]$. If Υ is not connected, we may introduce an artificial λ_{n+1} variable to the constraint, and append it $T \leftarrow T \cup \{n+1\}$ to each set $T \in \mathcal{T}$. The corresponding edge set $\Upsilon' = [d]^2$ is now connected, and we can simply impose that $\lambda_{n+1} \leqslant 0$ to recover our original constraint.

First, we observe that $b^{i,j} \cdot c^{i,j} = 0$, and so as $\mathcal{L}$ is two-dimensional, $M(b^{i,j}; \mathcal{L})$ is the hyperplane spanned by $c^{i,j}$. Furthermore, $\Upsilon = \{\, \{i,j\} \in [d]^2 \mid T^i \cap T^j \neq \varnothing \,\} \subseteq \Upsilon' = [d]^2$, and so (5.7a) will recover all the inequalities in (3.2a). It just remains to show that any inequality given by $\{i,j\} \in \Upsilon' \backslash \Upsilon$ is valid for $Q(\mathcal{P}(\mathcal{T}), H)$. To see this, consider any $(\lambda, z) = (\mathbf{e}^w, h^u) \in \operatorname{Em}(\mathcal{P}(\mathcal{T}), H)$; that is, $w \in T^u$. Then

$\sum_{v=1}^{n} \min_{s:v\in T^s}\{b^{i,j}\cdot h^s\}\lambda_v = \min_{s:w\in T^s}\{b^{i,j}\cdot h^s\} \leqslant b^{i,j}\cdot h^u$, as $b^{i,j}\cdot h^u$ is one of the terms appearing in the minimization. A similar argument holds for the other side of the constraint. $\square$

This result implies a quadratic $\mathcal{O}(d^2)$ upper bound on the number of general inequality constraints needed to construct an ideal MIP-with-holes formulations for *any* combinatorial disjunctive constraint, *regardless* of the choice of encoding. This is in sharp contrast to the traditional MIP setting, where binary encodings can–and typically do–lead to an exponential number of facets [131].

Furthermore, this can be strengthened to an $\mathcal{O}(d)$ upper bound on the number of general inequality constraints when we use the moment curve encoding.

Proposition 25. *Take $\mathcal{T} = (T^i \subseteq V)_{i=1}^d$. Then $(\lambda, z) \in Q(\mathcal{P}(\mathcal{T}), H_d^{\mathrm{mc}})$ if and only if*

$$\sum_{v\in V} \min_{s:v\in T^s}\{s(t-s)\}\lambda_v \leqslant tz_1 - z_2 \leqslant \sum_{v\in V} \max_{s:v\in T^s}\{s(t-s)\}\lambda_v \quad \forall t \in [\![3, 2d-1]\!] \tag{5.8a}$$

$$(\lambda, z) \in \Delta^V \times \mathbb{R}^2. \tag{5.8b}$$

Proof. Take any $\{i,j\} \in [d]^2$. Observe that

$$c^{i,j} \equiv h^j - h^i = (j-i, j^2 - i^2) = (j-i)\cdot(1, i+j),$$

and that $3 \leqslant i+j \leqslant 2d-1$. Therefore, for each $\{i,j\} \in [d]^2$, there is some $t \in [\![3, 2d-1]\!]$ and some $\alpha > 0$ such that $c^{i,j} = \alpha\cdot(1,t)$. Therefore, our representation here is equivalent to that in Proposition 24, up to constant nonzero scalings of some of the inequalities. $\square$

As a concrete example, consider the grid triangulation in Figure 5-4. The sets $\mathcal{T} = (T^i)_{i=1}^8$ correspond to each of the triangles, where

$$T^1 = \{1,2,4\},\quad T^2 = \{5,6,8\},\quad T^3 = \{3,5,6\},\quad T^4 = \{4,5,7\},$$
$$T^5 = \{5,7,8\},\quad T^6 = \{2,3,5\},\quad T^7 = \{2,4,5\},\quad T^8 = \{6,8,9\}.$$

Then a description for $Q(\mathcal{P}(\mathcal{T}), H_8^{\text{mc}})$ is:

$$\begin{aligned}
4\lambda_1 + 4\lambda_2 + 6\lambda_3 + 4\lambda_4 + 6\lambda_5 + 6\lambda_6 + 4\lambda_7 + 6\lambda_8 - 24\lambda_9 &\geqslant 5z_1 - z_2 \\
6\lambda_1 + 6\lambda_2 + 12\lambda_3 + 12\lambda_4 + 12\lambda_5 + 12\lambda_6 + 12\lambda_7 + 10\lambda_8 - 8\lambda_9 &\geqslant 7z_1 - z_2 \\
7\lambda_1 + 7\lambda_2 + 12\lambda_3 + 7\lambda_4 + 7\lambda_5 + 0\lambda_6 + 15\lambda_7 + 0\lambda_8 + 0\lambda_9 &\leqslant 8z_1 - z_2 \\
8\lambda_1 + 8\lambda_2 + 18\lambda_3 + 8\lambda_4 + 14\lambda_5 + 8\lambda_6 + 20\lambda_7 + 8\lambda_8 + 8\lambda_9 &\leqslant 9z_1 - z_2 \\
8\lambda_1 + 18\lambda_2 + 18\lambda_3 + 20\lambda_4 + 20\lambda_5 + 18\lambda_6 + 20\lambda_7 + 20\lambda_8 + 8\lambda_9 &\geqslant 9z_1 - z_2 \\
9\lambda_1 + 9\lambda_2 + 21\lambda_3 + 9\lambda_4 + 16\lambda_5 + 16\lambda_6 + 24\lambda_7 + 16\lambda_8 + 16\lambda_9 &\leqslant 10z_1 - z_2 \\
10\lambda_1 + 30\lambda_2 + 30\lambda_3 + 28\lambda_4 + 30\lambda_5 + 24\lambda_6 + 30\lambda_7 + 30\lambda_8 + 24\lambda_9 &\geqslant 11z_1 - z_2 \\
12\lambda_1 + 42\lambda_2 + 42\lambda_3 + 42\lambda_4 + 42\lambda_5 + 40\lambda_6 + 40\lambda_7 + 40\lambda_8 + 40\lambda_9 &\geqslant 13z_1 - z_2 \\
(\lambda, z) &\in \Delta^9 \times \mathbb{R}^2.
\end{aligned}$$

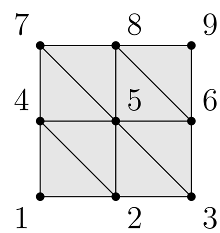


Figure 5-4: A grid triangulation with $d = 8$ triangles. The nodes, or vertices for the triangles, are numbered.

The construction of Proposition 25 gives these 8 facet-inducing general inequality constraints, along with 16 others that are valid but not facet-inducing for $Q(\mathcal{P}(\mathcal{T}), H_8^{\text{mc}})$, and therefore are not necessary. In contrast, any ideal binary MIP formulation requires three integer variables and at least 9 general inequality constraints [66].

We can also apply Proposition 25 to produce an ideal MIP-with-holes formulation for the SOS2 constraint with a linear number of inequality constraints and only two integer variables. Additionally, as we see in Figure 5-5, the moment curve branching scheme of Chapter 5.2.2 induces the same hereditarily sharp, incremental branching of the `Inc` formulation we observed in Chapter 4.1.2. Indeed, branching on the moment curve formulation (5.8) by imposing $\Psi_d(1, k)$ (resp. $\Psi_d(k+1, d)$) is equivalent to variable branching with the `Inc` formulation on $z_k \leqslant 0$ (resp. $z_k \geqslant 1$).

As we see in Figure 5-6, even variable branching on the moment curve formulation (5.8) for the SOS2 constraint induces the incremental branching behavior, though we do lose hereditary sharpness. Qualitatively, the branching is slightly weaker than the branching of both the `ZZI` and the `Inc` formulation as observed in Chapter 4.1.2, and there will be pathological cases in which variable branching of this type is not possible. However, it is noteworthy that, in nearly every case, we can induce incremental branching using only two integer variables and variable branching.

5.3.4 A very small formulation for the SOS2 constraint

We can further sharpen our general results in Proposition 24–which apply to any combinatorial disjunctive constraint–if we take advantage of structure in $\mathcal{T}$ and choose an encoding tailored for a particular constraint. For example, the exotic encoding H_d^{ex} was specifically designed for the SOS2 constraint $\mathcal{T}_d^{\text{SOS2}} \equiv (\{i, i+1\})_{i=1}^d$. Recall that the difference directions we need to compute for Theorem 9 are all differences between adjacent codes: $C = \{h^{i+1} - h^i\}_{i=1}^{d-1}$. Referring back to Figure 5-2, we see that these difference directions are all axis-aligned in the plane. That is, there are only two hyperplanes we need to consider, which in turn leads to a formulation with very few inequality constraints.

Proposition 26. *Take $d = 4r$ for some $r \in \mathbb{N}$, and label $H_d^{\text{ex}} = (h^i)_{i=1}^d \subset \mathbb{R}^2$. Then $(\lambda, y) \in Q(\mathcal{P}(\mathcal{T}_d^{\mathsf{SOS2}}), H_d^{\text{ex}})$ if and only if*

$$h_k^1 \lambda_1 + \sum_{i=2}^d \min\{h_k^{i-1}, h_k^i\} \lambda_i + h_k^d \lambda_{d+1} \leqslant z_k \qquad \forall k \in [\![2]\!] \tag{5.9a}$$

$$h_k^1 \lambda_1 + \sum_{i=2}^d \max\{h_k^{i-1}, h_k^i\} \lambda_i + h_k^d \lambda_{d+1} \geqslant z_k \qquad \forall k \in [\![2]\!] \tag{5.9b}$$

$$(\lambda, z) \in \Delta^{d+1} \times \mathbb{R}^2. \tag{5.9c}$$

Proof. Apply Theorem 9, after observing that $C = \{h^{i+1} - h^i\}_{i=1}^{d-1} \subseteq \{\pm \mathbf{e}^1, \pm \mathbf{e}^2\}$, and so taking the hyperplanes given by the normal directions $b^1 = \mathbf{e}^1$ and $b^2 = \mathbf{e}^2$

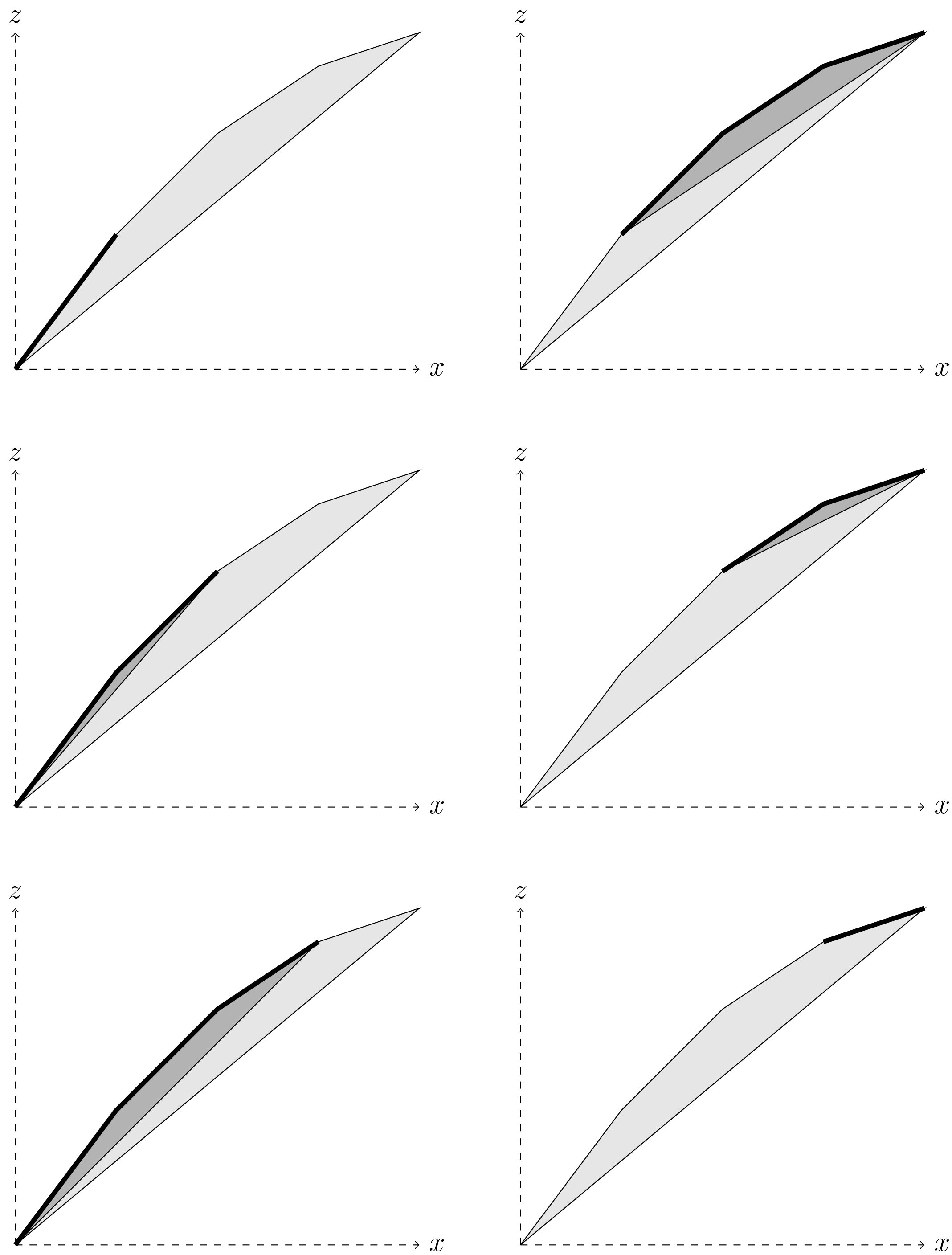


Figure 5-5: The LP relaxation of the moment curve formulation (5.8) applied to the the piecewise linear function (1.11) projected onto (x, y)-space, after **(Top Left)** branching on $\Psi_4(1,1)$, **(Top Right)** branching on $\Psi_4(2,4)$; **(Center Left)** branching on $\Psi_4(1,2)$, **(Center Right)** branching on $\Psi_4(3,4)$; **(Bottom Left)** branching on $\Psi_4(1,3)$, and **(Bottom Right)** branching on $\Psi_4(4,4)$.

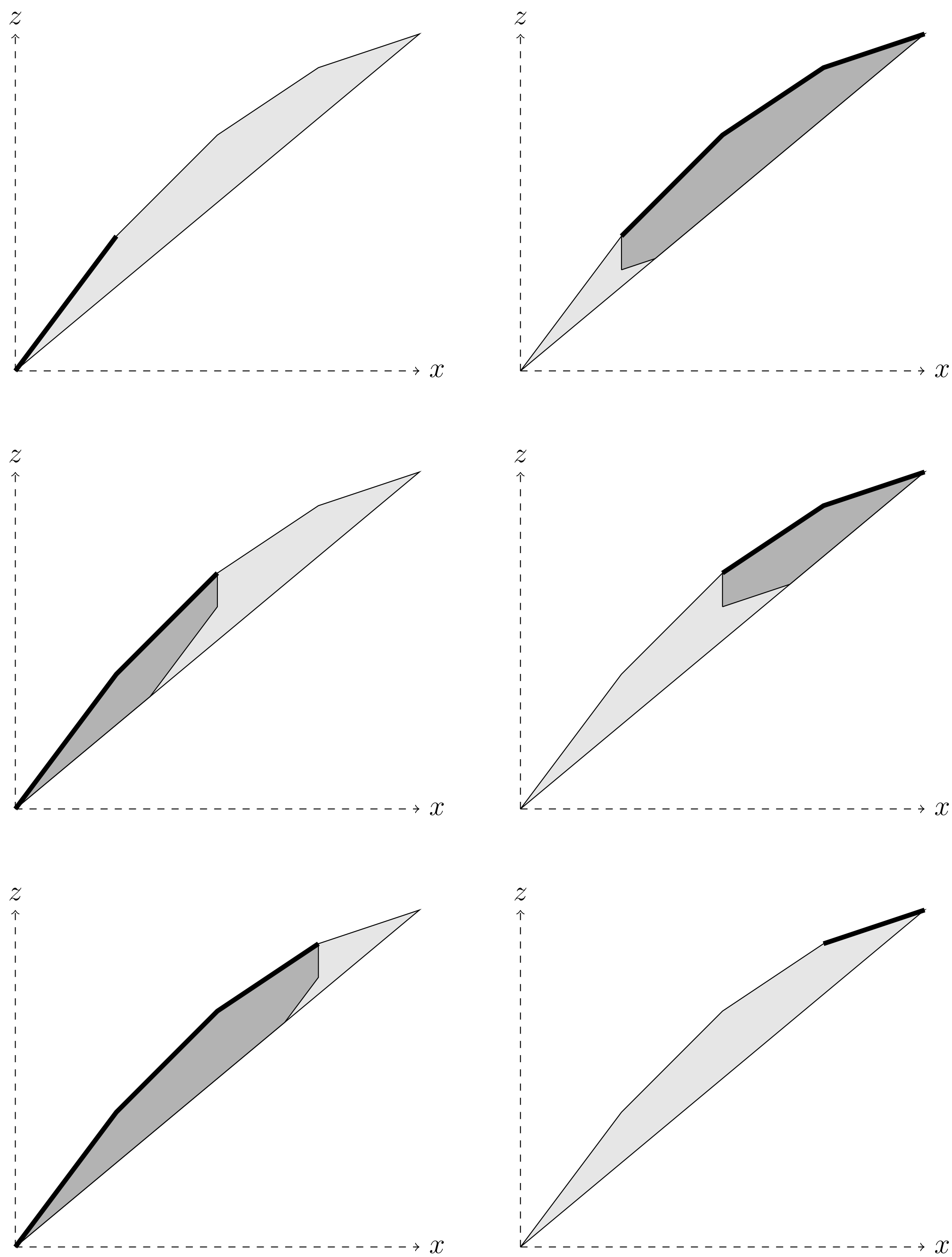


Figure 5-6: The LP relaxation of the moment curve formulation (5.8) projected onto (x, y)-space, after **(Top Left)** down-branching $z_1 \leqslant 1$, **(Top Right)** up-branching $z_1 \geqslant 2$; **(Center Left)** down-branching $z_1 \leqslant 2$, **(Center Right)** up-branching $z_1 \geqslant 3$; **(Bottom Left)** down-branching $z_1 \leqslant 3$, and **(Bottom Right)** up-branching $z_1 \geqslant 4$.

suffices. $\square$

As a concrete example, formulation (5.9) for the SOS2(17) constraint is

$$-4\lambda_1 - 4\lambda_2 + 4\lambda_3 - 3\lambda_4 - 3\lambda_5 - 3\lambda_6 + 3\lambda_7 - 2\lambda_8 - 2\lambda_9 +$$
$$-2\lambda_{10} + 2\lambda_{11} - 1\lambda_{12} - 1\lambda_{13} - 1\lambda_{14} + 1\lambda_{15} + 0\lambda_{16} + 0\lambda_{17} \leqslant z_1 \tag{5.10a}$$
$$-4\lambda_1 + 4\lambda_2 + 4\lambda_3 + 4\lambda_4 - 3\lambda_5 + 3\lambda_6 + 3\lambda_7 + 3\lambda_8 - 2\lambda_9 +$$
$$2\lambda_{10} + 2\lambda_{11} + 2\lambda_{12} - 1\lambda_{13} + 1\lambda_{14} + 1\lambda_{15} + 1\lambda_{16} + 0\lambda_{17} \geqslant z_1 \tag{5.10b}$$
$$0\lambda_1 + 0\lambda_2 + 0\lambda_3 + 4\lambda_4 - 4\lambda_5 - 4\lambda_6 - 4\lambda_7 + 7\lambda_8 - 7\lambda_9 +$$
$$-7\lambda_{10} - 7\lambda_{11} + 9\lambda_{12} - 9\lambda_{13} - 9\lambda_{14} - 9\lambda_{15} + 10\lambda_{16} + 10\lambda_{17} \leqslant z_2 \tag{5.10c}$$
$$0\lambda_1 + 0\lambda_2 + 4\lambda_3 + 4\lambda_4 + 4\lambda_5 - 4\lambda_6 + 7\lambda_7 + 7\lambda_8 + 7\lambda_9 +$$
$$-7\lambda_{10} + 9\lambda_{11} + 9\lambda_{12} + 9\lambda_{13} - 9\lambda_{14} + 10\lambda_{15} + 10\lambda_{16} + 10\lambda_{17} \leqslant z_2 \tag{5.10d}$$
$$(\lambda, z) \in \Delta^{17} \times \mathbb{R}^2. \tag{5.10e}$$

This MIP-with-holes formulation is ideal and uses only two integer variables, along with only four general integer inequality constraints. Indeed, this size is independent of d. We contrast this with the logarithmic formulations discussed in Chapter 3 (i.e. `Log`, `LogIB`, `ZZB`, and `ZZI`), which are also ideal but require 4 integer variables and 8 general inequality constraints for SOS2(17). Moreover, this size will grow logarithmically in d.

In Figure 5-7, we explore the branching behavior of very small MIP-with-holes formulation (5.9) when $d = 4$ by depicting the four possibilities of variable branching on z_1. We observe that the branching is neither hereditarily sharp, nor does it have the incremental branching property. However, we note that its branching appears to be only slightly inferior than that of the `Log`/`LogIB` formulations, which enjoy excellent computational results due to their size and strength, in spite of their branching behavior. In contrast, the MIP-with-holes formulation (5.9) is just as strong and substantially smaller.

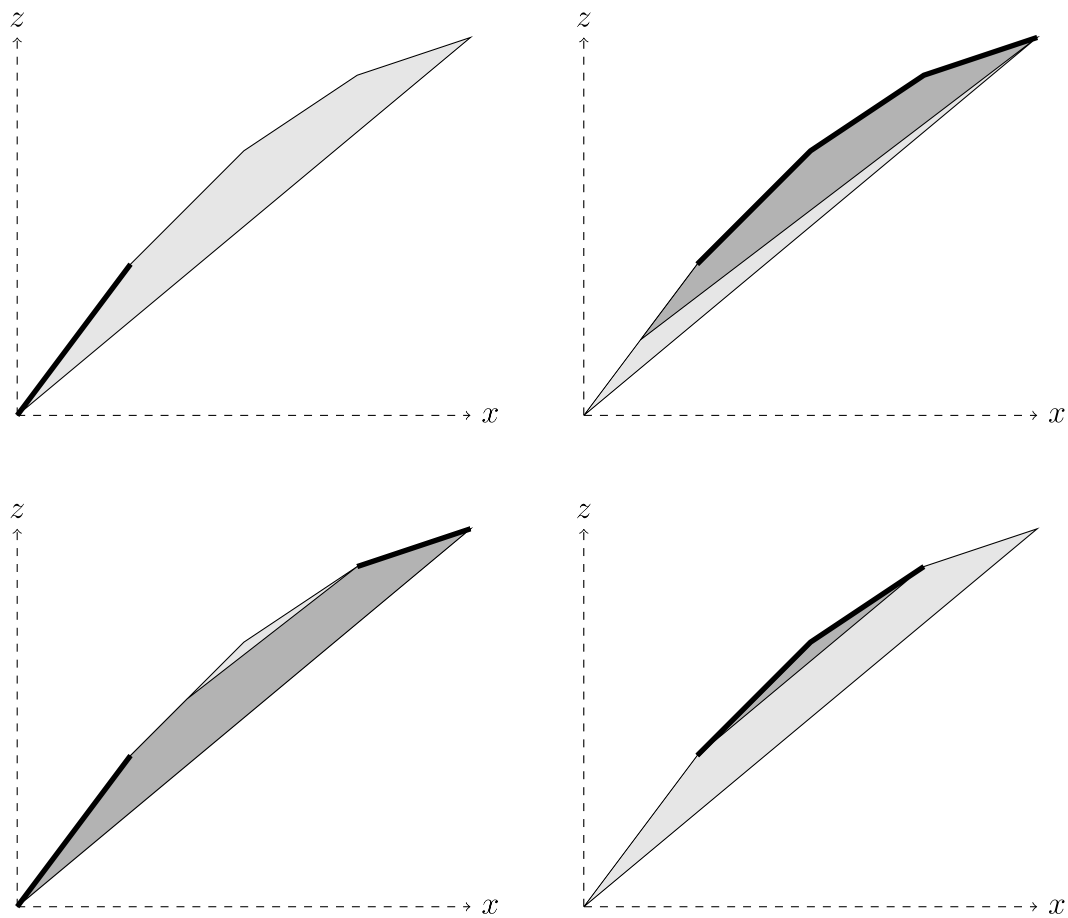


Figure 5-7: The LP relaxation of the very small SOS2 MIP-with-holes formulation (5.9) projected onto (x, y)-space, after **(Top Left)** down-branching $z_1 \leqslant -1$, **(Top Right)** up-branching $z_1 \geqslant 0$; **(Center Left)** down-branching $z_1 \leqslant 0$, and **(Center Right)** up-branching $z_1 \geqslant 1$.

5.3.5 A very small formulation for the annulus

We close the thesis by presenting a very small MIP-with-holes formulation for the annulus. This formulation also uses the exotic encoding H_d^{ex}, and requires only a constant number of integer variables and general inequality constraints.

Proposition 27. *Take $H_d^{\text{ex}} = (h^i)_{i=1}^d$, along with $h^{d+1} \equiv h^1$ for notational convenience. Then $(\lambda, z) \in Q(\mathcal{P}(\mathcal{T}_d^{\text{ann}}), H_d^{\text{ex}})$ if and only if*

$$\sum_{i=1}^{d} \min\{h_k^i, h_k^{i+1}\}(\lambda_{2i-1} + \lambda_{2i}) \leqslant z_k \qquad \forall k \in [\![2]\!] \tag{5.11a}$$

$$\sum_{i=1}^{d} \max\{h_k^i, h_k^{i+1}\}(\lambda_{2i-1} + \lambda_{2i}) \geqslant z_k \qquad \forall k \in [\![2]\!] \tag{5.11b}$$

$$\sum_{i=1}^{d} \min\{w \cdot h^i, w \cdot h^{i+1}\}(\lambda_{2i-1} + \lambda_{2i}) \leqslant w \cdot z \tag{5.11c}$$

$$\sum_{i=1}^{d} \max\{w \cdot h^i, w \cdot h^{i+1}\}(\lambda_{2i-1} + \lambda_{2i}) \geqslant w \cdot z \tag{5.11d}$$

$$(\lambda, z) \in \Delta^{2d} \times \mathbb{R}^2, \tag{5.11e}$$

where $w = (h_2^d - h_2^1, h_1^1 - h_1^d)$.

Proof. As $\Upsilon = \{i, i+1\}_{i=1}^{d-1} \cup \{1, d\}$, then $C \subseteq \{\pm \mathbf{e}^1, \pm \mathbf{e}^2, h^d - h^1\}$, and the result immediately follows from Theorem 9 as $w \cdot (h^d - h^1) = 0$. $\square$

In Figure 5-8, we depict the branching behavior of formulation (5.11) for the annulus with $d = 8$ quadrilateral pieces.

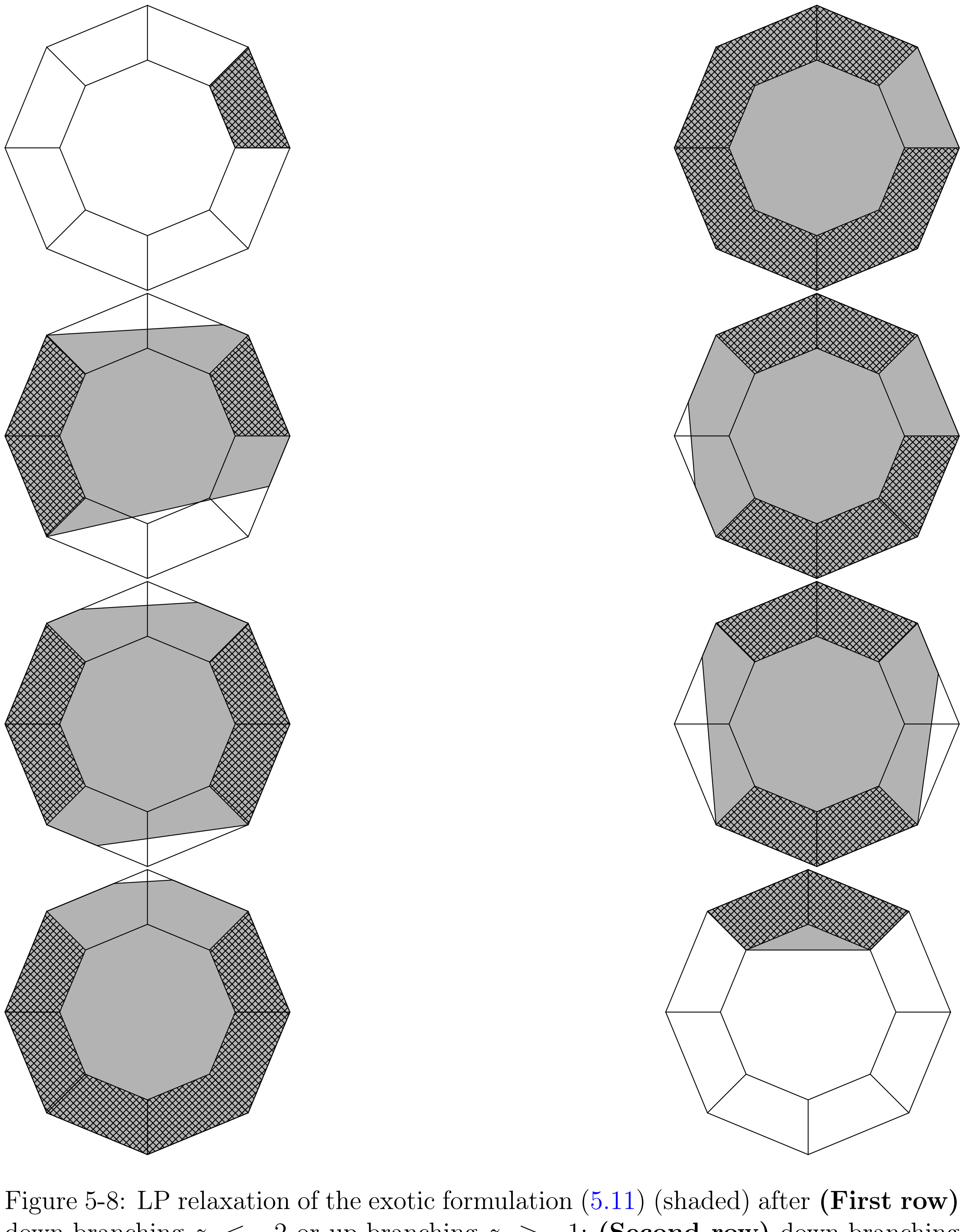


Figure 5-8: LP relaxation of the exotic formulation (5.11) (shaded) after **(First row)** down-branching $z_1 \leqslant -2$ or up-branching $z_1 \geqslant -1$; **(Second row)** down-branching $z_1 \leqslant -1$ or up-branching $z_1 \geqslant 0$; **(Third row)** down-branching $z_1 \leqslant 0$ or up-branching $z_1 \geqslant 1$; or **(Last row)** down-branching $z_1 \leqslant 1$ or up-branching $z_1 \geqslant 2$. The quadrilaterals that are feasible for each subproblem are crosshatched.

Appendix A

The logarithmic formulation of Misener et al. [106] is not ideal

We show that the logarithmic formulation (16) from Misener et al. [106] is not, in general, ideal. Using their notation, we take $N_P = 3$, $x^L = y^L = 0$, and $x^U = y^U = 3$

(and so $a = 1$). Then formulation (16) is

$$\lambda_1 + 2\lambda_2 \leqslant x \tag{A.1a}$$

$$x \leqslant 1 + \lambda_1 + 2\lambda_2 \tag{A.1b}$$

$$1 + \lambda_1 + 2\lambda_2 \leqslant 3 \tag{A.1c}$$

$$\Delta y_1 \leqslant 3\lambda_1 \tag{A.1d}$$

$$\Delta y_2 \leqslant 3\lambda_2 \tag{A.1e}$$

$$\Delta y_1 = y - s_1 \tag{A.1f}$$

$$\Delta y_2 = y - s_2 \tag{A.1g}$$

$$s_1 \leqslant 3(1 - \lambda_1) \tag{A.1h}$$

$$s_2 \leqslant 3(1 - \lambda_2) \tag{A.1i}$$

$$z \geqslant \Delta y_1 + 2\Delta y_2 \tag{A.1j}$$

$$z \geqslant 3x + (y - 3) + (\Delta y_1 - 3\lambda_1) + 2(\Delta y_2 - 3\lambda_2) \tag{A.1k}$$

$$z \leqslant y + \Delta y_1 + 2\Delta y_2 \tag{A.1l}$$

$$z \leqslant 3x + (\Delta y_1 - 3\lambda_1) + 2(\Delta y_2 - 3\lambda_2) \tag{A.1m}$$

$$\lambda \in \{0, 1\}^2 \tag{A.1n}$$

$$\Delta y \in [0, 3]^2 \tag{A.1o}$$

$$s \in [0, 3]^2 \tag{A.1p}$$

$$(x, y) \in [0, 3] \times [0, 3]. \tag{A.1q}$$

The feasible point for the relaxation $x = 3$, $y = 3$, $z = 9$, $\lambda = (1, 0.5)$, $\Delta y = (3, 1.5)$, and $s = (0, 1.5)$ is a fractional extreme point, showing that the formulation is not ideal. Indeed, it satisfies at equality the set of linear independent constraints of the relaxation given by $x \leqslant 3$, $y \leqslant 3$, $\lambda_1 \leqslant 1$, $\Delta y_1 \leqslant 3$, $s_1 \geqslant 0$, (A.1b), (A.1e), (A.1g) and (A.1k).

Appendix B

Constructing a pairwise IB-representable partition of the plane

We outline the operations that can transform an arbitrary partition of $\Omega \subset \mathbb{R}^2$ into one that conforms with the conditions of Theorem 2. In particular, the partition will have no "internal vertices," and no independent sets T in $H^c_{\mathcal{T}}$ with $|T| > 2$, where T is infeasible for $\mathrm{CDC}(\mathcal{T})$. We provide a sketch of the argument, accompanied by pictures in Figure B-1.

First, we see how we can remove internal vertices. Consider some $v \in V$ that is internal to $P^{i'}$ (i.e. $v \in P^{i'}$ and $v \notin \mathrm{ext}(P^{i'})$). We may split $P^{i'}$ in two along $P^{i',1} \cup P^{i',2} = P^{i'}$ in such a way that $P^{i',1} \cup P^{i',2} \cup \left(\bigcup_{i \in [\![k]\!] \setminus \{i'\}} P^i \right)$, without introducing any additional internal vertices. See the Left column of Figure B-1 for an illustration. Repeating this procedure will yield a valid polygonal partition of Ω with no internal vertices. We note that this implies that, for any bounded nonconvex region $\Omega \subset \mathbb{R}^2$, there exists a polyhedral complex whose union is Ω.

Secondly, we turn our attention to minimal infeasible sets $T \subseteq V$ with cardinality $|T| = 3$. If $\mathrm{Conv}(T) \subseteq \Omega$, we may append an additional set $P = \mathrm{Conv}(T)$ to our partition, and augment $P^i \leftarrow \mathrm{Conv}(\mathrm{ext}(P) \backslash S)$ for each i, where $S = \{ v \in V \mid v \in \mathrm{int}(\mathrm{Conv}(T)) \}$. See the Center column of Figure B-1.

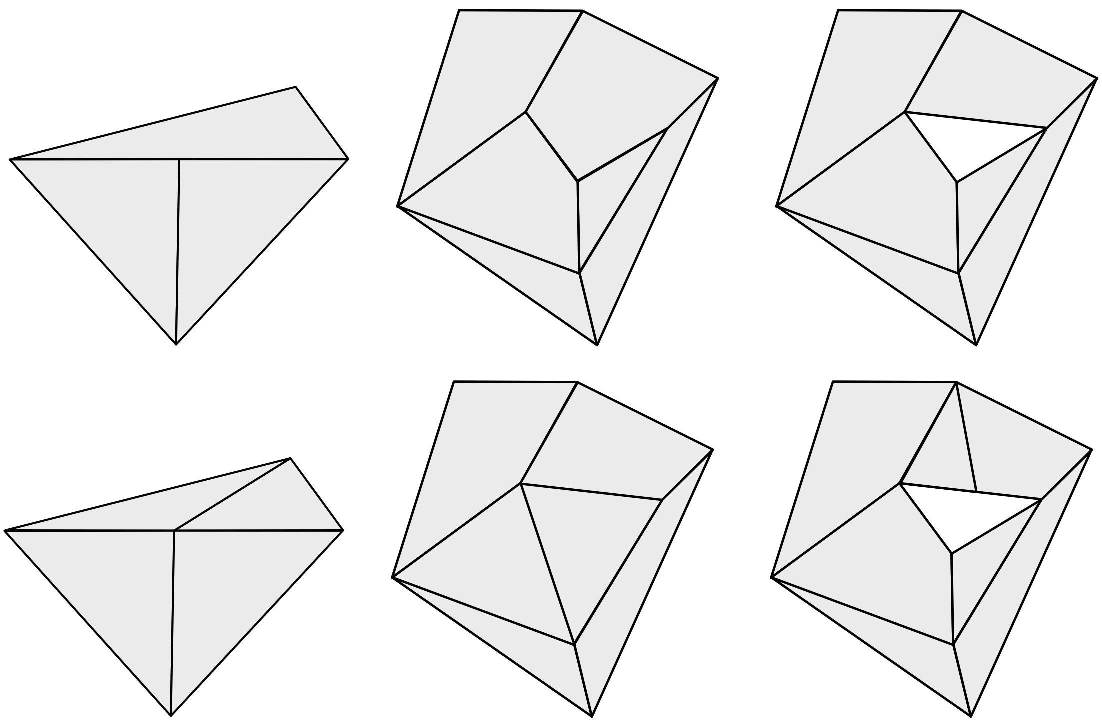


Figure B-1: The joining or splitting which transform a polygonal partition into one adhering to the conditions in Theorem 2 (before on Top, after on Bottom): **(Left)** splitting along an internal vertex, **(Center)** filling in a minimal infeasible set T of cardinality three with $\mathrm{Conv}(T) \subset \Omega$, and **(Right)** introducing an artificial vertex to remove a minimal infeasible set T of cardinality three with $\mathrm{Conv}(T) \not\subset \Omega$.

If $\mathrm{Conv}(T) \nsubseteq \Omega$, label $T = \{t^1, t^2, t^3\}$. It must be that one pair in T, w.l.o.g. (t^1, t^2), induces an edge on the boundary of Ω. Artificially introduce a vertex $v \in \mathrm{int}(\mathrm{Conv}(\{t^1, t^2\}))$. Now $\{t^1, t^2\}$ is an infeasible set, and so T is no longer a minimal infeasible set w.r.t. inclusion. See the Right column of Figure B-1.

Appendix C

8-segment piecewise linear function formulation branching

Consider the univariate piecewise linear function $f : [0, 8] \to \mathbb{R}$ given by

$$f(x) = \begin{cases} 8x & 0 \leqslant x \leqslant 1 \\ 7x + 1 & 1 \leqslant x \leqslant 2 \\ 6x + 3 & 2 \leqslant x \leqslant 3 \\ 5x + 6 & 3 \leqslant x \leqslant 4 \\ 4x + 10 & 4 \leqslant x \leqslant 5 \\ 3x + 15 & 5 \leqslant x \leqslant 6 \\ 2x + 21 & 6 \leqslant x \leqslant 7 \\ x + 28 & 7 \leqslant x \leqslant 8. \end{cases} \tag{C.1}$$

The corresponding `LogIB`/`Log` formulation is

$$x = \lambda_2 + 2\lambda_3 + 3\lambda_4 + 4\lambda_5 + 5\lambda_6 + 6\lambda_7 + 7\lambda_8 + 8\lambda_9 \tag{C.2a}$$

$$y = 8\lambda_2 + 15\lambda_3 + 21\lambda_4 + 26\lambda_5 + 30\lambda_6 + 33\lambda_7 + 35\lambda_8 + 36\lambda_9 \tag{C.2b}$$

$$z_1 \geqslant \lambda_3 + \lambda_7 \tag{C.2c}$$

$$z_1 \leqslant \lambda_2 + \lambda_3 + \lambda_4 + \lambda_6 + \lambda_7 + \lambda_8 \tag{C.2d}$$

$$z_2 \geqslant \lambda_4 + \lambda_5 + \lambda_6 \tag{C.2e}$$

$$z_2 \leqslant \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 \tag{C.2f}$$

$$z_3 \geqslant \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 \tag{C.2g}$$

$$z_3 \leqslant \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 \tag{C.2h}$$

$$(\lambda, z) \in \Delta^9 \times \{0,1\}^3, \tag{C.2i}$$

and the corresponding `ZZI` formulation is

$$x = \lambda_2 + 2\lambda_3 + 3\lambda_4 + 4\lambda_5 + 5\lambda_6 + 6\lambda_7 + 7\lambda_8 + 8\lambda_9 \tag{C.3a}$$

$$y = 8\lambda_2 + 15\lambda_3 + 21\lambda_4 + 26\lambda_5 + 30\lambda_6 + 33\lambda_7 + 35\lambda_8 + 36\lambda_9 \tag{C.3b}$$

$$z_1 \geqslant \lambda_3 + \lambda_4 + 2\lambda_5 + 2\lambda_6 + 3\lambda_7 + 3\lambda_8 + 4\lambda_9 \tag{C.3c}$$

$$z_1 \leqslant \lambda_2 + \lambda_3 + 2\lambda_4 + 2\lambda_5 + 3\lambda_6 + 3\lambda_7 + 4\lambda_8 + 4\lambda_9 \tag{C.3d}$$

$$z_2 \geqslant \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + 2\lambda_8 + 2\lambda_9 \tag{C.3e}$$

$$z_2 \leqslant \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + 2\lambda_7 + 2\lambda_8 + 2\lambda_9 \tag{C.3f}$$

$$z_3 \geqslant \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 \tag{C.3g}$$

$$z_3 \leqslant \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 \tag{C.3h}$$

$$(\lambda, z) \in \Delta^9 \times \{0,1,2,3,4\} \times \{0,1,2\} \times \{0,1\} \tag{C.3i}$$

In Table C.1, we show statistics for the relaxations of the both. We observe that the `ZZI` formulation yields more balanced branching, with the volume and strengthened proportion more equal between the resulting two branches.

Statistic	`Log` 0 ↓	`Log` 1 ↑	`ZZI` 0 ↓	`ZZI` 1 ↑	`ZZI` 1 ↓	`ZZI` 2 ↑	`ZZI` 2 ↓	`ZZI` 3 ↑	`ZZI` 3 ↓	`ZZI` 4 ↑
Volume	41	17	0	38.5	11.5	27	27	11.5	38.5	0
Str. Prop.	0	1	1	0.25	0.75	0.5	0.5	0.75	0.25	1

Table C.1: Metrics for each possible branching decision on z_1 for `Log` and `ZZI` applied to (C.1).

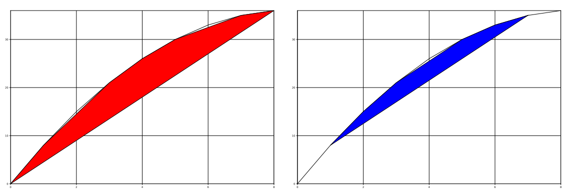


Figure C-1: Feasible region in the (x, y)-space for the `Log` formulation (C.2) after: **(Left)** down-branching $z_1 \leqslant 0$, and **(Right)** up-branching $z_1 \geqslant 1$.

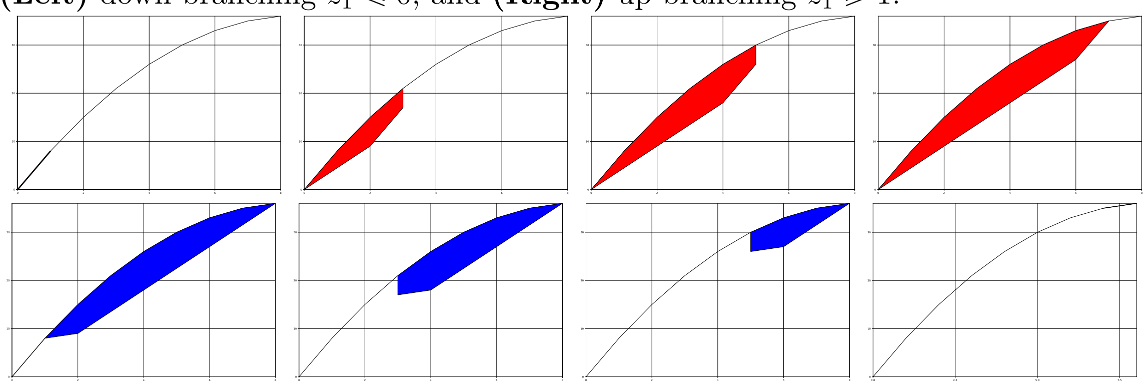


Figure C-2: Feasible region in the (x, y)-space for the `ZZI` formulation (C.3) after: **(Top first column)** down-branching on $z_1 \leqslant 0$, **(Bottom first column)** up-branching on $z_1 \geqslant 1$; **(Top second column)** down-branching on $z_1 \leqslant 1$, **(Bottom second column)** up-branching on $z_1 \geqslant 2$; **(Top third column)** down-branching on $z_1 \leqslant 2$, **(Bottom third column)** up-branching on $z_1 \geqslant 3$; **(Top fourth column)** down-branching on $z_1 \leqslant 3$, and **(Bottom fourth column)** up-branching on $z_1 \geqslant 4$.

Appendix D

Additional computational results with Gurobi

d	Metric	MC	CC	SOS2	Inc	DLog	Log	LogIB	ZZB	ZZI
28	Mean (s)	124.6	245.8	1784.8	31.5	27.1	19.8	**16.3**	19.7	17.0
	Std	192.9	321.4	151.9	16.1	15.8	15.3	**6.8**	11.3	9.3
	Win	0	0	0	0	5	16	**38**	11	30
	Fail	**0**	2	99	**0**	**0**	**0**	**0**	**0**	**0**
59	Mean (s)	619.4	901.2	1800.0	87.3	23.9	27.4	26.3	24.7	**20.9**
	Std	560.3	683.5	-	53.6	19.7	**11.8**	14.1	16.5	16.1
	Win	0	0	0	0	10	9	20	7	**54**
	Fail	12	27	100	**0**	**0**	**0**	**0**	**0**	**0**

Table D.1: Computational results with Gurobi for univariate transportation problems on large networks with non powers-of-two segments.

- See Table D.1 for univariate computational results on large networks (cf. Table 4.3).
- See Table D.2 for bivariate computational results (cf. Table 4.6).

κ	Metric	MC	CC	DLog	6-Stencil + Log	ZZB	ZZI
4	Mean (s)	1.1	1.8	0.7	**0.3**	**0.3**	**0.3**
	Std	0.8	1.6	0.6	**0.1**	**0.1**	**0.1**
	Win	0	0	0	**43**	20	37
	Fail	**0**	**0**	**0**	**0**	**0**	**0**
8	Mean (s)	13.0	54.9	12.4	**2.1**	2.3	**2.1**
	Std	12.5	79.9	14.8	2.2	2.1	**1.9**
	Win	0	0	0	**52**	19	29
	Fail	**0**	**0**	**0**	**0**	**0**	**0**
16	Mean (s)	440.8	1154.9	266.7	**16.0**	18.7	16.2
	Std	560.9	724.3	438.3	21.1	20.6	**18.8**
	Win	0	0	0	**45**	12	23
	Fail	6	39	3	**0**	**0**	**0**
32	Mean (s)	1521.6	1799.0	1291.1	**111.6**	129.0	121.0
	Std	515.6	8.8	599.8	**145.8**	156.6	163.6
	Win	0	0	0	**48**	10	22
	Fail	56	79	38	**0**	**0**	**0**

Table D.2: Computational results with Gurobi for bivariate transportation problems on grids of size $\kappa = d_1 = d_2$.

Bibliography

[1] Tobias Achterberg, Timo Berthold, Thorsten Koch, and Kati Wolter. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: 5th International Conference, CPAIOR 2008 Paris, France, May 20-23, 2008 Proceedings*, chapter Constraint integer programming: A new approach to integrate CP and MIP, pages 6–20. Springer Berlin Heidelberg, 2008.

[2] Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.

[3] Oswin Aichholzer, Franz Aurenhammer, Ferran Hurtado, and Hannes Krasser. Towards compatible triangulations. *Theoretical Computer Science*, 296(1):3–13, 2003.

[4] Kent Andersen, Gérard Cornuéjols, and Yanjun Li. Split closure and intersection cuts. *Mathematical Programming, Series A*, 102:457–493, 2005.

[5] Ross Anderson and Ondrej Sykora. Solving argmax for a neural network with MIP, and related optimization problems. Unpublished (personal communication), 2018.

[6] Ioannis Androulakis and Costas D. Maranas. ÎśBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7(4):337–363, 1995.

[7] J. A. Appleget and R. K. Wood. Explicit-constraint branching for solving mixed-integer programs. In M. Laguna and J. L. González , editors, *Computing tools for modeling, optimization, and simulation: interfaces in computer science and operations research*, volume 12 of *Operations research / computer science interfaces series*, pages 245–261. Kluwer, 2000.

[8] Krzysztof R. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.

[9] Jorg Bader, Robert Hildebrand, Robert Weismantel, and Rico Zenklusen. Mixed integer reformulations of integer programs and the affine TU-dimension of a matrix. *Mathematical Programming*, 2017.

[10] Anantaram Balakrishnan and Stephen C. Graves. A composite algorithm for a concave-cost network flow problem. *Networks*, 19:175–202, 1989.

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

[12] Egon Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:3–44, 1998.

[13] E. M. L. Beale and J. A. Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In J. Lawrence, editor, *OR 69: Proceedings of the Fifth International Conference on Operational Research*, pages 447–454. Tavistock Publications, 1970.

[14] Nicholas Beaumont. An algorithm for disjunctive programs. *European Journal of Operational Research*, 48(3):362–371, October 1990.

[15] John Saunders Bellingham. Coordination and control of uav fleets using mixed-integer linear programming. Master's thesis, Massachusetts Institute of Technology, 2002.

[16] Maria Lorena Bergamini, Pio Aguirre, and Ignacio Grossmann. Logic-based outer approximation for globally optimal synthesis of process networks. *Computers and Chemical Engineering*, 29(9):1914–1933, August 2005.

[17] Maria Lorena Bergamini, Ignacio Grossmann, Nicolás Scenna, and Pio Aguirre. An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms. *Computers and Chemical Engineering*, 32(3):477–493, March 2008.

[18] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, July 2017.

[19] Dimitris Bertsimas and Angela King. An algorithmic approach to linear regression. *Operations Research*, 64(1):2–16, 2015.

[20] Dimitris Bertsimas and Romy Shioda. Algorithm for cardinality-constrained quadratic optimization. *Comput. Optim. Appl.*, 43:1–22, 2009.

[21] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[22] Daniel Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74:121–140, 1996.

[23] Bob Bixby. 1000x MIP tricks. Presentation at Bill Cunningham's 65th birthday, June 2012.

[24] R. Bixby and E. Rothberg. Progress in computational mixed integer programming—A look back from the other side of the tipping point. *Annals of Operations Research*, 149:37–41, 2007.

[25] Yuri Bogomolov, Samuel Fiorini, Aleksandr Maksimenko, and Kanstantsin Pashkovich. Small extended formulations for cyclic polytopes. *Discrete Comput Geom*, 53:809–816, 2015.

[26] Pierre Bonami, Michele Conforti, and Gerard Cornuejols. Cutting planes from two-term disjunctions. *Operations Research Letters*, 41(5):442–444, 2013.

[27] Pierre Bonami, Andrea Lodi, Andrea Tramontani, and Sven Wiese. Cutting planes from wide split disjunctions. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 99–110, Cham, 2017. Springer.

[28] Pedro M. Castro and João P. Teles. Comparison of global optimization algorithms for the design of water-using networks. *Computers and Chemical Engineering*, 52:249–261, May 2013.

[29] Marcel Celaya. Lattice-free polytopes with lots of vertices. https://people.math.gatech.edu/~jyu67/teaching/2015Fall7012/proj/Celaya.pdf, 2015.

[30] T.-J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha. Heuristics for cardinality constrained portfolio optimization. *Computers and Operations Research*, 27:1271–1302, 2000.

[31] Chih-Hong Cheng, Georg Nührenberg, and Harald Ruess. Maximum resilience of artificial neural networks. https://arxiv.org/abs/1705.01040, 2017.

[32] Andres Codas and Eduardo Camponogara. Mixed-integer linear optimization for optimal lift-gas allocation with well-separator routing. *European Journal of Operational Research*, 217(1):222–231, February 2012.

[33] Andres Codas, Sthener Campos, Eduardo Camponogara, Vidar Gunnerud, and Snjezana Sunjerga. Integrated production optimization of oil fields with pressure and routing constraints: The Urucu field. *Computers and Chemical Engineering*, 46:178–189, November 2012.

[34] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming*. Springer, 2014.

[35] Don Coppersmith and Jon Lee. Parsimonious binary-encoding in integer programming. *Discrete Optimization*, 2:190–200, 2005.

[36] Denis Cornaz and Jean Fonlupt. Chromatic characterization of biclique covers. *Discrete Mathematics*, 306:495–507, 2006.

[37] Yves Crama and Peter L Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.

[38] Keely L. Croxton, Bernard Gendron, and Thomas L. Magnanti. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science*, 49(9):1268–1273, September 2003.

[39] Keely L. Croxton, Bernard Gendron, and Thomas L. Magnanti. Variable disaggregation in network flow problems with piecewise linear costs. *Operations Research*, 55(1):146–157, January-February 2007.

[40] Bai Cui and Xu Andy Sun. A new voltage stability-constrained optimal power flow model: Sufficient condition, SOCP representation, and relaxation. *IEEE Transactions on Power Systems*, 2017.

[41] Claudia D'Ambrosio, Andrea Lodi, and Silvano Martello. Piecewise linear approximation of functions of two variables in MILP models. *Operations Research Letters*, 38(1):39–46, January 2010.

[42] George B. Dantzig. On the significance of solving linear programming problems with some integer variables. *Econometrica, Journal of the Econometric Society*, pages 30–44, 1960.

[43] I. R. de Farias Jr., E. L. Johnson, and G. L. Nemhauser. Branch-and-cut for combinatorial optimization problems without auxiliary binary variables. *The Knowledge Engineering Review*, 16:25–39, 2001.

[44] I. R. de Farias Jr., E. Kozyreff, R. Gupta, and M. Zhao. Branch-and-cut for separable piecewise linear optimization and intersection with semi-continuous constraints. *Mathematical Programming Computation*, 5(1):75–112, 2013.

[45] I.R. de Farias Jr., M. Zhao, and H. Zhao. A special ordered set approach for optimizing a discontinuous separable piecewise linear function. *Operations Research Letters*, 36(2):234–238, March 2008.

[46] Robin Deits and Russ Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. In *2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 279–286. IEEE, 2014.

[47] Robin Deits and Russ Tedrake. Efficient mixed-integer planning for UAVs in cluttered environments. In *IEEE International Conference on Robotics and Automation*, pages 42–49, 2015.

[48] Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[49] Matthew G. Earl and Raffaello D'Andrea. Iterative MILP methods for vehicle-control problems. *IEEE Transactions on Robotics*, 21(6):1158–1167, December 2005.

[50] Matteo Fischetti and Jason Jo. Deep neural networks as 0-1 mixed integer linear programs: A feasibility study. https://arxiv.org/abs/1712.06174, 2017.

[51] Peter C. Fishburn and Peter L. Hammer. Bipartite dimensions and bipartite degrees of graphs. *Discrete Mathematics*, 160:127–148, 1996.

[52] Christodoulos A. Floudas and Xiaoxia Lin. Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Annals of Operations Research*, 139:131–162, 2005.

[53] James Daniel Foster. *Mixed-integer quadratically-constrained programming, piecewise-linear approximation and error analysis with applications in power flow*. PhD thesis, The University of Newcastle, Australia, November 2013.

[54] L.R. Foulds, D. Haugland, and K. Jörnsten. A bilinear approach to the pooling problem. *Optimization*, 24(1-2):165–180, 1992.

[55] R. Fourer, D. M. Gay, and B. Kernighan. *AMPL: a mathematical programming language*. Springer-Verlang, 1989.

[56] Armin Fügenschuh, Christine Hayn, and Dennis Michaels. Mixed-integer linear methods for layout-optimization of screening systems in recovered paper production. *Optimization and Engineering*, 15:533–573, 2014.

[57] Michael R. Garey and David S. Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[58] Bjorn Geißler, Alexander Martin, Antonio Morsi, and Lars Schewe. *Using piecewise linear functions for solving MINLPs*, pages 287–314. Springer, 2012.

[59] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, volume 1. MIT Press Cambridge, 2016.

[60] T. Graf, P. Van Hentenryck, C. Pradelles-Lasserre, and L. Zimmer. Simulation of hybrid circuits in constraint logic programming. *Computers and Mathematics with Applications*, 20(9–10):45–56, 1990.

[61] Hermann Gruber and Markus Holzer. *Inapproximability of nondeterministic state and transition complexity assuming $P \neq NP$*, volume 4588 of *Lecture Notes in Computer Science*, pages 205–216. Springer US, 2007.

[62] Hassan Hijazi, Pierre Bonami, Gérard Cornuéjols, and Adam Ouorou. Mixed-integer nonlinear programs featuring "on/off" constraints. *Computational Optimization and Applications*, 52(2):537–558, 2012.

[63] Hassan Hijazi, Pierre Bonami, and Adam Ouorou. A note on linear on/off constraints. http://www.optimization-online.org/DB_FILE/2014/04/4309.pdf, 2014.

[64] Robert Hildebrand, Robert Weismantel, and Rico Zenklusen. Extension complexity lower bounds for mixed-integer extended formulations. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2017.

[65] John N. Hooker. Logic, optimization, and constraint programming. *INFORMS Journal on Computing*, 14(4):295–321, Fall 2002.

[66] Joey Huchette and Juan Pablo Vielma. Nonconvex piecewise linear functions: Advanced formulations and simple modeling tools. https://arxiv.org/abs/1708.00050, 2017.

[67] Joey Huchette and Juan Pablo Vielma. A combinatorial approach for small and strong formulations of disjunctive constraints. *Mathematics of Operations Research*, 2018.

[68] Jr. Ismael R. de Farias and George L. Nemhauser. A polyhedral study of the cardinality constrained knapsack problem. *Mathematical Programming*, 96(3):439–467, 2003.

[69] Rabih A. Jabr. Radial distribution load flow using conic programming. *IEEE Transactions on Power Systems*, 21(3):1458–1459, 2006.

[70] Joxan Jaffar and Michael J. Maher. Constraint logic programming: A survey. *The Journal of Logic Programming*, 20:503–581, 1994.

[71] R. G. Jeroslow and J. K. Lowe. Experimental results on the new techniques for integer programming formulations. *The Journal of the Operational Research Society*, 36(5):393–403, May 1985.

[72] R.G. Jeroslow and J.K. Lowe. Modelling with integer variables. *Mathematical Programming Study*, 22:167–184, 1984.

[73] Robert G. Jeroslow. Alternative formulations of mixed integer programs. *Annals of Operations Research*, 12:241–276, 1988.

[74] Michael Jünger, Thomas Liebling, Denis Naddef, George Nemhauser, William Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence Wolsey. *50 years of integer programming 1958-2008*. Springer, 2010.

[75] A. B. Keha, I. R. de Farias, and G. L. Nemhauser. Models for representing piecewise linear cost functions. *Operations Research Letters*, 32:44–48, 2004.

[76] A. B. Keha, I. R. de Farias, and G. L. Nemhauser. A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Operations Research*, 54:847–858, 2006.

[77] Ahmet B. Keha, Ismael R. de Farias Jr., and George L. Nemhauser. Models for representing piecewise linear cost functions. *Operations Research Letters*, 32(1):44–48, January 2004.

[78] Ahmet B. Keha, Ismael R. de Farias Jr., and George L. Nemhauser. A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Operations Research*, 54(5):847–858, September-October 2006.

[79] T. Kis. Lift-and-project for general two-term disjunctions. *Discrete Optimization*, 12:98–114, 2014.

[80] Thorsten Koch, Benjamin Hiller, Marc E. Pfetsch, and Lars Schewe, editors. *Evaluating Gas Network Capacities*. MOS-SIAM Series on Optimization. SIAM, 2015.

[81] Burak Kocuk, Santanu S. Dey, and X. Andy Sun. Strong SOCP relaxations for optimal power flow. *Operations Research*, 64(6):1177–1196, 2016.

[82] Scott Kolodziej, Pedro M. Castro, and Ignacio E. Grossmann. Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization*, 57:1039–1063, 2013.

[83] E. Kondili, C. C. Pantelides, and R. W. H. Sargent. A general algorithm for short-term scheduling of batch operations–I. MILP formulation. *Computers and Chemical Engineering*, 17(2):211–227, 1993.

[84] Harold W. Kuhn. Some combinatorial lemmas in topology. *IBM Journal of Research and Development*, 4(5):518–528, 1960.

[85] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, 2016.

[86] Ailsa Land and Alison Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[87] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[88] Jon Lee, Daphne Skipper, and Emily Speakman. Algorithmic and modeling insights via volumetric comparison of polyhedral relaxations. *Mathematical Programming*, 2018.

[89] Jon Lee and Dan Wilson. Polyhedral methods for piecewise-linear functions I: the lambda method. *Discrete Applied Mathematics*, 108:269–285, 2001.

[90] Haoxiang Liu and David Z.W. Wang. Global optimization method for network design problem with stochastic user equilibrium. *Transportation Research Part B: Methodological*, 72:20–39, February 2015.

[91] Jianfeng Liu, Michael Bynum, Anya Castillo, Jean-Paul Watson, and Carl D. Laird. A multitree approach for global solution of ACOPF problems using piecewise outer approximations. *Computers and Chemical Engineering*, 2017.

[92] Miles Lubin and Iain Dunning. Computing in operations research using Julia. *INFORMS Journal on Computing*, 27(2):238–248, Spring 2015.

[93] Miles Lubin, Ilias Zadik, and Juan Pablo Vielma. Mixed-integer convex representability. In Friedrich Eisenbrand, editor, *International Conference on Integer Programming and Combinatorial Optimization*, pages 392–404, Cham, 2017. Springer International Publishing.

[94] Miles Lubin, Ilias Zadik, and Juan Pablo Vielma. Regularity in mixed-integer convex representability. https://arxiv.org/abs/1706.05135, 2017.

[95] James Luedtke, Mahdi Namazifar, and Jeff Linderoth. Some results on the strength of relaxations of multilinear functions. *Mathematical Programming*, 136(2):325–351, 2012.

[96] Thomas L. Magnanti and Dan Stratila. Separable concave optimization approximately equals piecewise linear optimization. In Daniel Bienstock and George Nemhauser, editors, *Lecture Notes in Computer Science*, volume 3064, pages 234–243. Springer, 2004.

[97] D. Mahlke, A. Martin, and S. Moritz. A mixed integer approach for time-dependent gas network optimization. *Optimization Methods and Software*, 25(4):625–644, August 2010.

[98] Patrick Scott Mara. Triangulations for the cube. *Journal of Combinatorial Theory*, 20:170–177, 1976.

[99] Harry M. Markowitz and Alan S. Manne. On the solution of discrete programming problems. *Econometrica*, 25(1):84–110, January 1957.

[100] A. Martin, M. Moller, and S. Moritz. Mixed integer models for the stationary case of gas network optimization. *Mathematical Programming*, 105:563–582, 2006.

[101] Alexander Martin, Markus Möller, and Susanne Moritz. Mixed integer models for the stationary case of gas network optimization. *Mathematical Programming*, 105(2-3):563–582, February 2006.

[102] Garth McCormick. Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

[103] Daniel Mellinger, Alex Kushleyev, and Vijay Kumar. Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In *IEEE International Conference on Robotics and Automation*, pages 477–483, 2012.

[104] Ruth Misener and Christodoulos Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Mathematical Programming*, 136(1):155–182, December 2012.

[105] Ruth Misener, Chrysanthos E. Gounaris, and Christodoulos A. Floudas. Global optimization of gas lifting operations: A comparative study of piecewise linear formulations. *Industrial and Engineering Chemistry Research*, 48(13):6098–6104, 2009.

[106] Ruth Misener, Jeffrey P. Thompson, and Christodoulos A. Floudas. APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers and Chemical Engineering*, 35:876–892, 2011.

[107] Frank Muldoon. *Polyhedral approximations of quadratic semi-assignment problems, disjunctive programs, and base-2 expansions of integer variables*. PhD thesis, Clemson University, Clemson, SC, USA, 2012.

[108] Frank M. Muldoon, Warren P. Adams, and Hanif D. Sherali. Ideal representations of lexicographic orderings and base-2 expansions of integer variables. *Operations Research Letters*, 41:32–39, 2013.

[109] James Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae (Proceedings)*, 80(5):406–424, 1977.

[110] James Ostrowski, Jeff Linderoth, Fabrizio Rossi, and Stefano Smriglio. Orbital branching. *Mathematical Programming*, 126(1):147–178, 2011.

[111] Manfred Padberg. Approximating separable nonlinear functions via mixed zero-one programs. *Operations Research Letters*, 27:1–5, 2000.

[112] Panos Pardalos and Stephen Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization*, 1(1):15–22, March 1991.

[113] Ionela Prodan, Florin Stoican, Sorin Olaru, and Silviu-Iulian Niculescu. Enhancements on the hyperplanes arrangements in mixed-integer programming techniques. *Journal of Optimization Theory and Applications*, 154(2):549–572, 2012.

[114] Ionela Prodan, Florin Stoican, Sorin Olaru, and Silviu-Iulian Niculescu. *Mixed-Integer Representations in Control Design: Mathematical Foundations and Applications*. SpringerBriefs in Electrical and Computer Engineering. Springer, 2016.

[115] Ignacio Quesada and Ignacio E. Grossmann. A global optimization algorithm for linear fractional and bilinear programs. *Journal of Global Optimization*, 6(1):39–76, 1995.

[116] Steffen Rebennack. Computing tight bounds via piecewise linear functions through the example of circle cutting problems. *Mathematical Methods of Operations Research*, 84:3–57, 2016.

[117] Richard Rosenthal. *GAMS - A User's Guide*. GAMS Development Corporation, 2014.

[118] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. *Computer Scheduling of Public Transport*, pages 269–280, 1981.

[119] Nokolaos V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8:201–205, 1996.

[120] Carla Savage. A survey of combinatorial Gray codes. *SIAM Review*, 39(4):605–629, December 1997.

[121] Buser Say, Ga Wu, Yu Qing Zhou, and Scott Sanner. Nonlinear hybrid planning with deep net learned transition models and mixed-integer linear programming. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 750–756, 2017.

[122] Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. https://arxiv.org/abs/1711.02114, 2018.

[123] Hanif D. Sherali and Hongjie Wang. Global optimization of nonconvex factorable programming problems. *Mathematical Programming*, 89(3):459–478, 2001.

[124] Thiago L. Silva, Andres Codas, and Eduardo Camponogara. A computational analysis of convex combination models for multidimensional piecewise-linear approximation in oil production optimization. In *Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, pages 292–298, 2012.

[125] Thiago Lima Silva and Eduardo Camponogara. A computational analysis of multidimensional piecewise-linear models with applications to oil production optimization. *European Journal of Operational Research*, 232(3):630–642, February 2014.

[126] Vincent Tjeng and Russ Tedrake. Verifying neural networks with mixed integer programming. https://arxiv.org/abs/1711.07356, 2017.

[127] Michael J. Todd. Union Jack triangulations. *Fixed Points: Algorithms and Applications*, pages 315–336, 1977.

[128] J.A. Tomlin. A suggested extension of special ordered sets to non-separable nonconvex programming problems. *North-Holland Mathematics Studies*, 59:359–370, 1981.

[129] Juan Pablo Vielma. Embedding formulations and complexity for unions of polyhedra. https://arxiv.org/abs/1506.01417, 2015.

[130] Juan Pablo Vielma. Mixed integer linear programming formulation techniques. *SIAM Review*, 57(1):3–57, 2015.

[131] Juan Pablo Vielma. Embedding formulations and complexity for unions of polyhedra. *Management Science*, 2017.

[132] Juan Pablo Vielma, Shabbir Ahmed, and George Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

[133] Juan Pablo Vielma, Shabbir Ahmed, and George Nemhauser. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research*, 58(2):303–315, 2010.

[134] Juan Pablo Vielma, Ahmet B. Keha, and George L. Nemhauser. Nonconvex, lower semicontinuous piecewise linear optimization. *Discrete Optimization*, 5(2):467–488, May 2008.

[135] Juan Pablo Vielma and George Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128(1-2):49–72, 2011.

[136] Danan S. Wicaksono and Iftekhar A. Karimi. Piecewise MILP under- and overestimators for global optimisation of bilinear programs. *AIChE Journal*, 54(4):991–1008, 2008.

[137] Daniel L. Wilson. *Polyhedral methods for piecewise-linear functions*. PhD thesis, University of Kentucky, Lexington, Kentucky, 1998.

[138] Ga Wu, Buser Say, and Scott Sanner. Scalable planning with Tensorflow for hybrid nonlinear domains. In *Advances in Neural Information Processing Systems*, pages 6276–6286, 2017.

[139] Sercan Yildiz and Juan Pablo Vielma. Incremental and encoding formulations for mixed integer programming. *Operations Research Letters*, 41:654–658, 2013.

[140] Gunter Ziegler. *Lectures on Polytopes*. Springer, 2007.