

THE COMPLEXITY OF EXTENDED FORMULATIONS

A Dissertation
Presented to
The Academic Faculty

by

Arefin Huq

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology

August 2016

Copyright © 2016 Arefin Huq

THE COMPLEXITY OF EXTENDED FORMULATIONS

Approved by:

Dr. Lance Fortnow, Co-advisor
School of Computer Science
Georgia Institute of Technology

Dr. Greg Blekherman
School of Mathematics
Georgia Institute of Technology

Dr. Richard Lipton
School of Computer Science
Georgia Institute of Technology

Dr. Sebastian Pokutta, Co-advisor
School of Industrial and
Systems Engineering
Georgia Institute of Technology

Dr. Santosh Vempala
School of Computer Science
Georgia Institute of Technology

Date Approved: May 11, 2016

Tyger Tyger, burning bright,
In the forests of the night;
What immortal hand or eye,
Could frame thy fearful symmetry?
William Blake

To Mom and Dad.

ACKNOWLEDGMENTS

First I would like to thank my parents. I dedicate this work to my dad, the first PhD in the family, and to my mom, who may be the only person who never doubted I could get there too. I could fill another thesis with what you both have done for me and I would just be getting started. Thank you for everything.

I'm incredibly lucky to have had not one but two wonderful advisors, Lance Fortnow and Sebastian Pokutta. Lance: getting to be a complexity theorist has been a dream come true for me. Thank you for your support, your patience, and of course for bringing me down South. Sebastian: what started as a class to fill a minor requirement turned into the most productive research of my career. If I can remember just some of the many insights you've shared with me about research and life, I'll do very well.

I'd like to acknowledge the other members of my committee: Greg Blekherman, Dick Lipton, and Santosh Vempala. Thank you for your feedback and helpful discussions, many of which happened even before you agreed to serve on my committee. I'd also like to thank my coauthors and collaborators through the years, with special recognition for Gábor Braun: your guidance was more than once the critical piece I needed to move forward.

During these years of living on a graduate student's budget, I've counted my wealth in family and friends. I'd be foolish to try to do justice to all of you by name, so let me just say that your words and support have carried me through this journey, more than you may know.

One friend deserves special mention: Melanie Wertz. You were practically my third advisor. You've seen me through this journey from one side to the other with a patient ear, encouraging words, and a wisdom that is truly out of this world.

I would like to thank Northwestern University, where I started my PhD, and Georgia Tech, where I finished, for fostering a supportive environment where students can succeed. I also want to thank my past mentors: Yann LeCun, Davi Geiger, Juan Bello, and Bryan Pardo. And of course none of my work would have been possible without the organizations that funded my research, including my home institutions and the National Science Foundation. It's a privilege to be paid to advance human knowledge.

Finally, I would like to express my heartfelt love and appreciation for Nora, my wife and teammate, whom I had the great fortune to meet during my PhD program. Marrying you is the greatest accomplishment I've had in a long time. Someone once told me that the love of a good woman will change a man's life. It's true.

TABLE OF CONTENTS

Acknowledgments	v
List of Tables	x
List of Figures	xi
List of Symbols	xii
Summary	xiv
Chapter 1: Introduction	1
1.1 P vs NP, TSP, and extended formulations	2
1.2 Combinatorial optimization	2
1.3 Using geometry	5
1.4 Extended formulations	10
1.5 Symmetry	12
1.6 Extended formulations as a model of computation	18
1.7 Contribution	19
1.8 Related work	20
Chapter 2: Mathematical Background	24
2.1 Notation	24
2.2 Basic definitions from convex geometry	26
2.3 Polytopes	29
2.3.1 Projection	31
2.4 Slack Matrices	31
2.5 Linear, semidefinite, and copositive programming	34

Chapter 3: Symmetric SDP Extended Formulations	36
3.1 Symmetric SDP formulations	36
3.2 The symmetric factorization lemma	38
Chapter 4: A Symmetric SDP Lower Bound for Matching	43
4.1 The perfect matching problem	43
4.1.1 Symmetric functions on matchings are juntas	43
4.1.2 The matching polynomials	45
4.1.3 Deriving that symmetrized polynomials are constant	46
4.1.4 Low-degree certificates for matching ideal membership	48
4.1.5 The symmetric SDP lower bound	52
4.2 A note on the inapproximability ratio	54
Chapter 5: Small Symmetric CP Formulations	55
5.1 A symmetric CP extension for the correlation polytope	55
5.2 Connecting the cut and correlation polytopes	57
5.3 Maksimenko’s construction	58
5.4 Symmetric CNFs	59
5.5 A small symmetric CP formulation for matching	61
Chapter 6: A Small Symmetric CP Formulation for TSP	63
6.1 The construction	63
6.1.1 Variables	64
6.1.2 Encoding a permutation	65
6.1.3 Encoding cycles	65
6.1.4 From a permutation to a directed tour	66
6.1.5 From a directed tour to an undirected tour	66
6.1.6 Defining a group action	67
6.1.7 Putting it all together	67

Chapter 7: Conclusion	68
Appendix Chapter A: Permission to Reprint	69
References	71

LIST OF TABLES

1.1 Lower bounds for matching and TSP	22
---	----

LIST OF FIGURES

1.1	A graph with edge weights.	3
1.2	The feasible region of the fruit basket problem.	9
1.3	An extended formulation for the fruit basket problem.	11
1.4	180° rotational symmetry of the fruit basket problem.	13
1.5	Reflecting the extension of the fruit basket problem	15
1.6	90° rotational symmetry of the fruit basket problem	17

LIST OF SYMBOLS

- $A + B$ the Minkowski sum of sets A and B 25
- $a^\top b$ the inner product of vectors a and b 25
- $g \cdot x$ the action of group element g on x 25
- $\langle \mathcal{H} \rangle$ the vector space spanned by the polynomials \mathcal{H} 25
- $\langle \mathcal{H} \rangle_I$ the ideal generated by the polynomials \mathcal{H} 25
- \equiv congruence modulo an ideal 46
- $\simeq_{(\mathcal{P}, d)}$ congruence derivable from \mathcal{P} in degree d 46
- $[k]$ the numbers $\{1, \dots, k\}$ 24
- $\text{aff } X$ the affine hull of X 27
- A_n the alternating group on n letters 52
- CNF** conjunctive normal form 20
- $\text{cone } X$ the conic hull of X 28
- $\text{conv } X$ the convex hull of X 28
- \mathcal{C}_n the cone of $n \times n$ copositive matrices 33
- COR**(n) the correlation polytope 32
- CP** copositive program 12
- \mathcal{C}_n^* the cone of $n \times n$ completely positive matrices 33
- CUT**(n) the cut polytope 32
- $\deg p$ the degree of the polynomial p 25
- $\dim X$ the linear dimension of the set X 26
- e_i the i th basis vector 24
- K_n the complete graph on n vertices 43
- LP** linear program 10
- MaxCSP** maximum constraint satisfaction problem 21
- PM**(n) the perfect matching problem on K_n 43

\mathcal{P}_n the matching constraint polynomials 45
psd positive semidefinite 12
 \mathbb{S}_+^r the cone of real $n \times n$ symmetric psd matrices 25
 \mathbb{R} the set of real numbers 7, 24
 \mathbb{R}^d Euclidean space in d dimensions 24
 $(\mathbb{R}^d)^*$ dual space of \mathbb{R}^d 24
 relint P the relative interior of P 30
 $\mathbb{R}^{n \times n}$ the space of real $n \times n$ matrices 24
 \mathbb{R}_+ the nonnegative real numbers 24
 \mathbb{R}_{++} the strictly positive real numbers 24
 $\mathbb{R}[x]$ the set of polynomials in n real variables 25
SDP semidefinite program 12
 \mathbb{S}^n the set of real $n \times n$ symmetric matrices 24
 S_n the symmetric group on n letters 25
 span X the span of points in X 26
 \sqrt{M} the unique psd square root of the matrix M 38
 $\text{Tr}[AB]$ the Frobenius inner product of matrices A and B 25
TSP traveling salesperson problem 1
 vert P the set of vertices of P 30

SUMMARY

Combinatorial optimization plays a central role in complexity theory, operations research, and algorithms. Extended formulations give a powerful approach to solve combinatorial optimization problems: if one can find a concise geometric description of the possible solutions to a problem then one can use convex optimization to solve the problem quickly.

Many combinatorial optimization problems have a natural symmetry. In this work we explore the role of symmetry in extended formulations for combinatorial optimization, focusing on two well-known and extensively studied problems: the matching problem and the traveling salesperson problem.

In his groundbreaking work, Yannakakis [1991, 1988] showed that the matching problem does not have a small symmetric linear extended formulation. Rothvoß [2014] later showed that *any* linear extended formulation for matching, symmetric or not, must have exponential size. In light of this, we ask whether the matching problem has a small semidefinite extended formulation, since semidefinite programming generalizes linear programming. We show that the answer is no if the formulation is also required to be symmetric. Put simply, the matching problem does not have a small symmetric semidefinite extended formulation.

We next consider optimization over the copositive cone and its dual, the completely positive cone. Optimization in this setting is NP-hard. We present a general framework for producing compact symmetric copositive formulations for a large class of problems. We show that, in contrast to the semidefinite case, both the matching and traveling salesperson problems have small copositive formulations even if we require symmetry.

CHAPTER 1

INTRODUCTION

Combinatorial optimization plays a central role in complexity theory, operations research, and algorithms. In a combinatorial optimization problem one has a finite but typically large set of candidate solutions from which one wants the best solution based on some measure.

For example, consider trying to match medical students to residency programs. In this task the candidate solutions are all possible ways of assigning applicants to residencies, and we measure the quality of a solution by how well it satisfies the mutual preferences of applicants and hospitals.

For another example, consider planning the route of a delivery truck. Here the candidate solutions are all routes that visit each delivery location, and we measure the quality of a route by its total length.

Combinatorial optimization shows up everywhere. It features prominently in a wide range of modern scientific and commercial endeavors including biotechnology, engineering, manufacturing, and artificial intelligence. Algorithm designers naturally want to know how to solve such problems quickly, both in theory and in practice.

The residency program example is a version of the matching problem we will explore later. Even though the set of candidate solutions is exponentially large, the matching problem has practical, efficient algorithms. The route planning example is an instance of the traveling salesperson problem (TSP) which we will also explore later. The space of possible solutions to the TSP is also exponentially large, and in contrast to the matching problem, there is no known algorithm that quickly solves general instances of the TSP.

1.1 P vs NP, TSP, and extended formulations

Some combinatorial optimization problems, such as the TSP, seem to require an exhaustive search of exponentially many possibilities in the worst case. The famous P vs NP question asks, in essence, whether or not this is true. In fact, $P = NP$ if and only if there is an algorithm that solves every instance of the TSP in time that is only polynomial in the size of the instance.

The P vs NP problem was first formally stated by Cook [1971] and has attracted considerable attention ever since. Some researchers, including this author, consider the P vs NP problem to be the most important open problem in computer science and possibly all of mathematics. The P vs NP problem is one of the seven Millennium Problems selected by the Clay Mathematics Institute [2016], with a million dollar prize offered for its solution.

In the mid-1980's there was a series of attempts by Swart [1986] to show that $P = NP$ by giving a polynomial size linear program for the TSP. As we will see, such a construction, if correct, would have been a small linear extended formulation. According to accounts such as Trick [2009] and Lipton and Regan [2012], as reviewers found errors in the construction, and patches were introduced to fix those errors, the resulting linear program became increasingly complicated to analyze. In a breakthrough result, Yannakakis [1991, 1988] ended this line of inquiry by showing that any construction of this type was doomed to fail. In doing so, Yannakakis also founded the systematic study of extended formulations. We now describe the framework of combinatorial optimization and extended formulations in more detail.

1.2 Combinatorial optimization

Let us begin with a simple example. Consider the *graph* depicted in Figure 1.1 on page 3. We will call this graph G . The graph G has six *vertices* (also called *nodes*)

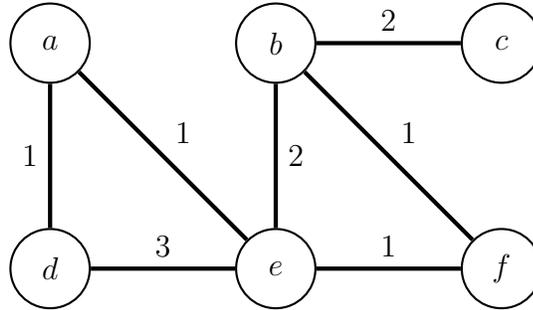


Figure 1.1: A graph with edge weights.

labeled a through f . Some pairs of vertices are connected by *edges*, and each edge is labeled with a number, called its *weight* (or *cost*). We will refer to edges by the vertices they connect, so for example the edge de is the edge with weight 3 that connects nodes d and e .

Graphs play a fundamental role in combinatorial optimization and algorithm design. They provide a rich and expressive notation for describing relationships among objects. For example, a graph could represent:

- an airline network, where each node is a city, edges are nonstop routes, and each edge weight is the cost of flying along that route;
- a molecule, where each node is an atom, edges are chemical bonds, and each edge weight is the strength of the bond;
- a social network, where each node is a person, edges are friendship relations, and each edge weight represents how much those friends communicate;
- an instance of the residency matching problem described earlier, where each node is a resident or hospital, edges are potential matches, and each edge weight is the desirability of including that particular match in the overall solution; or
- an instance of the route planning problem described earlier, where each node is a delivery location, edges are routes between locations, and each edge weight is the length of that route.

Algorithm designers often solve real-world problems by expressing them abstractly in terms of graphs and then applying graph algorithms to solve their problem. For example, the residency matching and route planning problems described earlier can both be expressed as graph problems. Since graphs are such a versatile tool, algorithm designers take great interest in knowing which graph problems can be solved easily.

The maximum matching problem

Let us look more closely at a particular graph problem, the *maximum matching problem*. A *matching* in a graph is a collection of edges such that no two edges share a common vertex. The vertices corresponding to any edge in the matching are said to be *matched*. For example, in the graph G , the set $\{ae, bf\}$ is a matching that matches four vertices in total, whereas the set $\{ad, ae\}$ is not a matching because the vertex a occurs twice. In a graph with edge weights, the weight of a matching is simply the sum of the weights of its edges. For example, the matching $\{ae, bf\}$ has a total weight of 2.

A *perfect matching* is a matching that covers every vertex. In order for a graph to be able to have a perfect matching it must have an even number of nodes. In the graph G , the set $\{ad, bc, ef\}$ is a perfect matching, and in fact the only one.

In the *maximum matching problem* the goal is to find a matching in an edge-weighted graph that has the highest total weight. In the *maximum perfect matching problem*, the goal is to find a perfect matching that has the highest total weight. For graph G the maximum perfect matching (indeed the only perfect matching) is $\{ad, bc, ef\}$, with a weight of 4. If we do not require a perfect matching, the maximum matching in G is $\{bc, de\}$, with a weight of 5.

We now review some standard terminology for combinatorial optimization problems. Recall that in a combinatorial optimization problem we seek the best solution, according to some measure, from a set of possible solutions. The possible solutions to

a problem are called the *feasible solutions*. In the example of maximum matching, the feasible solutions are all sets of edges that are matchings. The criterion that measures the quality of a solution is the *objective function* or simply the *objective*, which is a function that maps solutions to real numbers. In the example of maximum matching, the objective function is the sum of the weights of the edges in a matching.

In a *maximization problem* one seeks a feasible solution that maximizes the objective function, whereas in a *minimization problem* one seeks a feasible solution that minimizes the objective function. The maximum matching problem is indeed a maximization problem, since we seek the set of edges with the highest weighted sum. When we are speaking generically about an optimization problem (either a maximization or a minimization problem) we simply refer to *optimizing* the objective function.

The graph G is small enough that it is possible to list all its matchings and find the best one quickly. In a general combinatorial optimization problem the set of feasible solutions is always finite but typically very large. For example, a graph with only 40 nodes can have over 3×10^{23} perfect matchings, and that number grows exponentially with the size of the graph. A brute-force approach that evaluates all possible matchings quickly becomes intractable. This is the essence of the difficulty in combinatorial optimization: evaluating the quality of any given feasible solution is easy, but it is simply not possible to try all feasible solutions to find the best one. In order to have any hope of solving a typical combinatorial optimization problem, some other approach is needed.

1.3 Using geometry

One possible improvement over exhaustive search is inspired by convex geometry. Let us continue with the example of maximum matchings on the graph G . Finding a matching means selecting a subset of edges of G . Let's associate a *decision variable* with each edge of G that indicates whether we include that edge in our matching.

For example, we'll associate the variable x_{ad} with the edge ad , and we'll set x_{ad} to 1 if we include ad in our matching, and set it to 0 otherwise. Altogether we'll have seven such variables for the seven edges of G . If we pick an ordering of variables, say

$$x_{ad}, x_{ae}, x_{bc}, x_{be}, x_{bf}, x_{de}, x_{ef},$$

then any subset of edges of G (whether it is a matching or not) can be represented by a vector with seven entries, one for each variable, and with each entry equal to either 0 or 1. We refer to a vector whose entries are all either 0 or 1 as a *0/1 vector*.

For example, the edge set $\{ad, bc, ef\}$, corresponding to the unique perfect matching in G , is represented by the vector

$$(1, 0, 1, 0, 0, 0, 1),$$

while the edge set $\{bc, de\}$, corresponding to the maximum weight matching in G , is represented by the vector

$$(0, 0, 1, 0, 0, 1, 0).$$

Every feasible solution (that is, every matching in G) has a representation as a vector of this form, with seven entries each equal to either 0 or 1. On the other hand, some 0/1 vectors do not correspond to matchings. For example the vector

$$(1, 1, 0, 0, 0, 0, 0),$$

corresponding to the edge set $\{ad, ae\}$, does not represent a feasible solution since that edge set is not a matching. If the variable x_{ad} equals 1 then the variable x_{ae} must be 0 in order for those variables to encode a matching. This illustrates the fact that the variables are *correlated*: information about one variable can tell us about other variables.

Having mapped each feasible solution to a 0/1 vector, we now make the connection to geometry by viewing these vectors as points in seven-dimensional coordinate space, or in other words as elements of \mathbb{R}^7 . (Recall that \mathbb{R} denotes the real number line and that \mathbb{R}^2 denotes the Cartesian product of the real line with itself, also known as the coordinate plane.)

From now on we identify each feasible solution (each matching) with its corresponding point in space. We view the entire collection of points corresponding to matchings as defining the corners of a *polyhedron*. The set of all points in this polyhedron, including the corner points, form the *convex hull* of the points corresponding to matchings of G . We will refer to the convex hull of the points corresponding to feasible solutions as the *feasible region*.

We also represent the edge weights of G as a seven-dimensional vector we call c :

$$c = (1, 1, 2, 2, 1, 3, 1).$$

If x is an element of \mathbb{R}^7 that corresponds to a matching of G , then the weight of x , which is the sum of the weights of its edges, is given by

$$\text{weight}(x) = c_1x_1 + c_2x_2 + \dots + c_7x_7.$$

The key point is that the weight function (the objective) is a linear function of the coordinates of matchings. It is a fact of convex geometry that optimizing a linear function over a finite set of points is equivalent to optimizing the same function over its convex hull. If the finite set of points is very large but the convex hull has a compact geometric description, in a precise sense we will define, then it may be possible to use convex optimization to optimize the objective more quickly than an exhaustive search over the finite set. To apply the approach we have just described, we need:

1. a mapping from feasible solutions to points in some geometric space,

2. a representation of the objective as a linear function in this space, and
3. a compact description of the feasible region.

For the problems we will consider, namely the matching problem and the TSP, there are natural choices for the first two items, and the big question is whether one can have the last item, a compact description of the feasible region.

In this work we will study extended formulations as a way to get a compact description of the feasible region of a combinatorial optimization problem. In order to develop intuition for extended formulations we'll need an optimization problem that's easier to think about than the matching problem.

The fruit basket problem

Even though G is small, the feasible region of matchings in G is an object in seven-dimensional space, which is hard to visualize. To illustrate the idea of extended formulations we turn to an even simpler toy example, which we will refer to here as the fruit basket problem.

Imagine that we have three apples and three bananas available to make a fruit basket. Each apple is worth an amount a in the basket and each banana is worth b . Either a or b can be negative. The goal is to make a fruit basket with maximum value, subject to the constraint that we cannot pick an extreme amount (0 or 3) of both fruits. For example, if we pick 0 or 3 apples then we must pick 1 or 2 bananas. This somewhat contrived constraint models the correlation of variables we saw in the matching example: picking a value for one decision variable can restrict options for other decision variables.

Since there are two decision variables, x and y , the feasible region for this problem is a two-dimensional object, which we've drawn as the shaded region in Figure 1.2 on page 9. As can be seen, the feasible region is in fact an octagon, albeit not a regular octagon. The eight corners of the octagon are feasible solutions where exactly one of

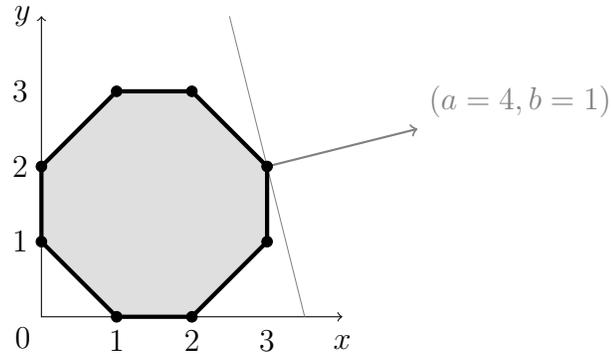


Figure 1.2: The feasible region of the fruit basket problem.

the two variables is set to an extreme value. The interior of the octagon contains four feasible solutions (not shown) where neither variable has an extreme value.

The figure also shows the solution to the fruit basket problem for the particular case $a = 4$ and $b = 1$. In this case the optimal choice is three apples and two bananas, represented by the point $(3, 2)$. The gray vector pointing out from the point $(3, 2)$ shows the direction in which the value of the basket increases for this choice of a and b . The thin gray line perpendicular to the vector shows a *level set* for the objective function, which is a set of points that have the same objective value. From this line it's easy to see that no other point in the feasible region has an objective value as high as that of the point $(3, 2)$.

The feasible region of the fruit basket problem can be described by eight linear inequalities, one for each side of the octagon:

$$x \geq 0, y \geq 0, x \leq 3, y \leq 3,$$

$$x + y \geq 1,$$

$$x + y \leq 5,$$

$$x - y \geq -2, \text{ and}$$

$$x - y \leq 2.$$

The objective is the linear function

$$f(x, y) = ax + by.$$

Since the fruit basket problem can be expressed as the optimization of a linear objective over a feasible region defined by linear inequalities, it is a *linear program (LP)*, which means it can be solved efficiently. Since the linear constraints exactly describe the convex hull of the feasible solutions, solving the linear program will give the exact optimum value of the original combinatorial problem.

Notice that the choice of costs a and b are encoded in the objective function and do not affect the feasible region. This means that different instances of the fruit basket problem have the same feasible region and differ only in the direction in which we wish to optimize. Here we see an appealing aspect of the geometric approach to optimization: if we can express the feasible region of a combinatorial optimization problem compactly, we can reuse that description for any instance of the problem; the particulars of each instance are encoded solely in the objective.

1.4 Extended formulations

The linear program for the fruit basket problem has eight inequalities. Is it possible to do better? As shown in Figure 1.3 on page 11, the answer is yes. The figure shows a three-dimensional object with six sides, which can be thought of as a tall rectangular prism that has been deformed by stretching the top face from left to right, and the bottom face from front to back, resulting in four trapezoidal faces around the sides.

If we shine a light on this object from directly above, it will cast an octagonal shadow on the ground, as depicted in the figure. In other words, if we *project* the object onto the xy plane, we obtain the octagon that is the feasible region of the fruit basket problem. We will refer to any higher dimensional object that can be projected

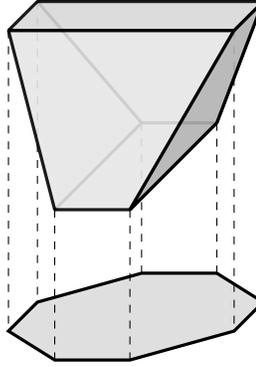


Figure 1.3: An extended formulation for the fruit basket problem.

exactly onto the octagon as an *extension* of the octagon, and refer to the octagon as the *projection* of any such extension.

Since the extension of the octagon shown in the figure has six flat sides it can be described by six inequalities in three variables. Since these six inequalities describe an extension of the octagon we will say they form an *extended formulation* of the octagon. If we are being precise we should distinguish between the geometric object (the extension) and its algebraic representation in terms of inequalities (the extended formulation), however we will typically use these terms interchangeably.

Since the extended formulation for the octagon uses linear inequalities we say that it forms a *linear extended formulation* of the fruit basket problem. The *size* of a linear extended formulation is the number of inequalities in its description. We do not consider the number of variables, the number of equality constraints, or the size of the coefficients in the inequalities. Although those numbers can matter in other contexts, for our purposes the number of inequalities turns out to be a more accurate measure of the geometric complexity of the object.

Using the number of inequalities as a measure of size, we would say that the original formulation of the fruit basket problem as a linear program in two variables has a size of eight, whereas the extended formulation in three dimensions has a size of six. One might wonder how far this idea can be carried. It turns out that there are objects described in terms of linear inequalities that have linear extended formulations

of exponentially smaller size; for an example consult Goemans [2015].

It is exactly this potential of exponential savings in problem description size that spurred on the study of extended formulations, since if the feasible region of a combinatorial optimization problem with exponentially many solutions can be described with exponentially fewer inequalities, it may be possible to solve the problem quickly with linear programming, as Swart attempted with the TSP. We now know that many problems, including the TSP, do not have an exponentially smaller description using linear formulations; see Section 1.8 for more discussion.

Linear programming is a special case of a general technique known as *conic programming*. In conic programming the feasible region of an optimization problem is expressed as the *affine slice* of a *closed convex cone* of a certain dimension. The power of a conic program (and the difficulty of solving it) lies in the type of cone that is used and in the dimension of the cone. Examples of cones are the *nonnegative cone*, the *positive semidefinite (psd) cone*, and the *copositive cone*.

The feasible region of any linear program can be expressed as the affine slice of the nonnegative cone of a certain dimension, where that dimension is equal to the number of inequalities in the linear program. *Semidefinite programs (SDPs)* generalize linear programs but are still solvable efficiently. Here the feasible region of a program is an affine slice of the psd cone, and the size of the program is the dimension of the psd cone used in its formulation. *Copositive programs (CPs)* generalize both linear and semidefinite programs. Here the feasible region is an affine slice of the copositive cone. Unlike linear and semidefinite programming, copositive programming is not believed to be efficiently solvable in general.

1.5 Symmetry

Many combinatorial problems, including matching and TSP, have natural symmetries. Generally speaking, a problem is symmetric if transformations like rotations and

reflections leave the feasible region unchanged. For example, the octagon that is the feasible region of the fruit basket problem has several symmetries: rotation by 90° about its center, and reflection about a horizontal line, vertical line, or 45° diagonal line through its center, sloping upwards or downwards. Since this octagon is not regular, it is not symmetric with respect to a 45° rotation about its center, as a regular octagon would be.

If we construct an extended formulation for a problem that has symmetries, we may find that the extended formulation respects those symmetries, or it may not. If an extended formulation respects the symmetries of the region it projects to, it is a *symmetric extended formulation*.

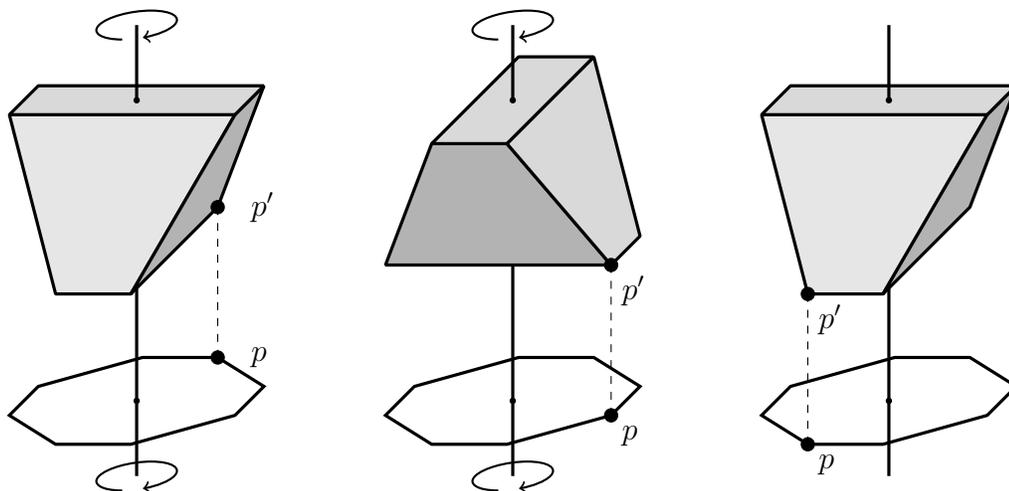


Figure 1.4: 180° rotational symmetry of the fruit basket problem.

For example, if we look at Figure 1.4 on page 13 we see that if we rotate the octagon by 180° about its center, and rotate the extension about this same axis, then by comparing the first and third diagrams in the figure we see that both the octagon and the extension are left unchanged, because both are symmetric with respect to a 180° rotation about that axis. We would say that the extension respects this particular symmetry.

What's required here is not just that the extension is unchanged by the rotation, but that each point in the extension is kept "in sync" with the point it projects to. What we mean formally by "in sync" is that the transformation that is applied to the extension must *commute* with the projection operation.

To understand what this means, consider the trajectory of the point p in the figure, as well as that of the point p' that projects to p . If we compare the first diagram with the last diagram, we see not only that the transformation has kept the octagon and the extension unchanged, but that the point p' still projects to p . We say that the transformation commutes with the projection because if we start from the first diagram and first project p' to p and then apply the rotation, we get the same result (that is, ending up at the rotated point p) as if we first apply the rotation to the extension and then project p' onto the plane of the octagon in the last diagram.

Having seen that this extension respects the 180° symmetry of the octagon, we can ask whether this extension respects all the symmetries of the octagon. It turns out there is a subtlety in the notion of "respecting symmetry." To understand this subtlety let's first examine Figure 1.5 on page 15. In this figure we show the effect of reflecting the extension through a plane that is parallel to the octagon and passes through the midpoint of the extension. This plane is depicted in the second diagram on the top row of the figure, and the subsequent diagrams show snapshots of the extension while the reflection is being applied. The first diagram on the bottom row, for example, shows the extension halfway through the reflection, at which point the entire extension is lying in the plane of reflection. The effect of reflecting the extension through this plane is similar to the effect of rotating the extension by 90° , except that if we follow the trajectory of the point p' during the transformation, we see that it moves only in a line parallel to the z axis, and that its projection p remains unchanged.

Now let's consider what happens to the extension of the fruit basket problem

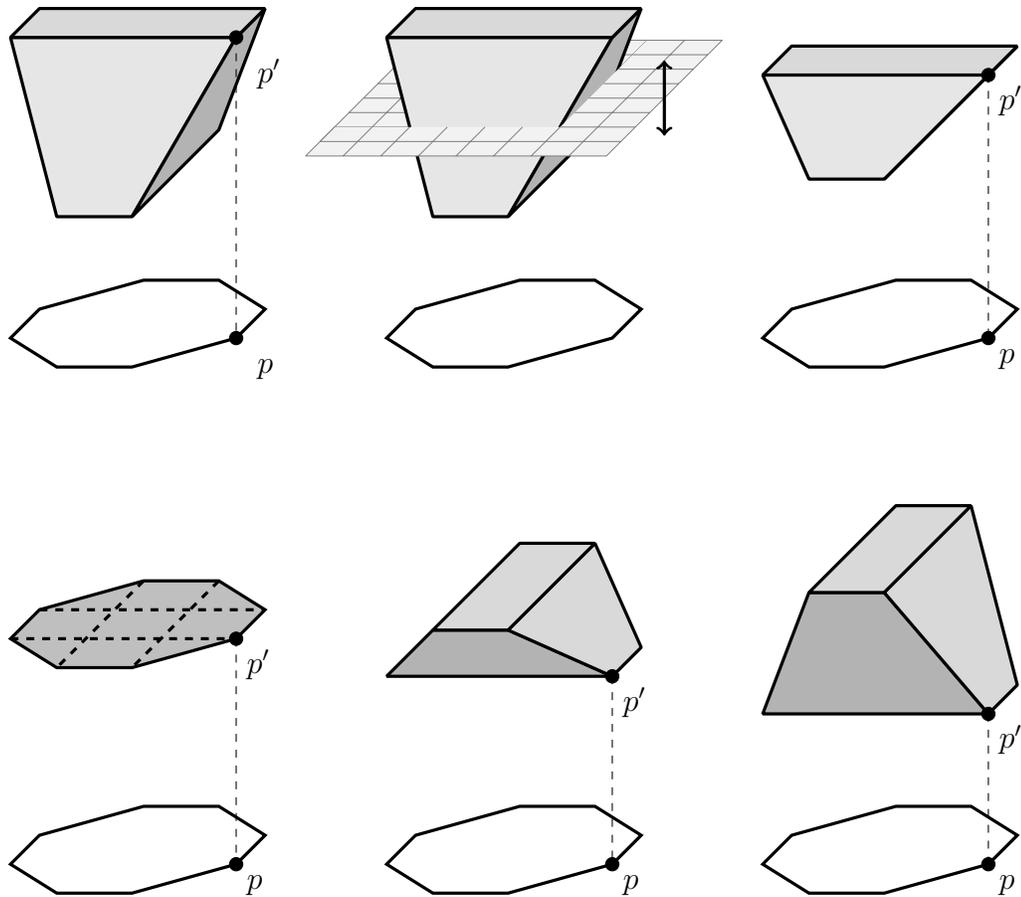


Figure 1.5: Reflecting the extended formulation of the fruit basket problem through a plane parallel to the octagon.

when we rotate the feasible region by 90° about its center, as shown in Figure 1.6 on page 17. In the top row we see that when we rotate the octagon 90° clockwise about its axis of symmetry and do the same to the extension, the extension ends up in a different orientation, since it does not have 90° rotational symmetry about this axis.

This would appear to show that the extension does not respect this particular symmetry of the octagon, however this is not the end of the story. If we now reflect the extension through a plane that is parallel to the octagon and passes through the midpoint of the extension, as we described in Figure 1.5 on page 15, the result is that the extension is now in the same orientation as it was at the start. This process is depicted in the bottom row of Figure 1.6 on page 17. Because we were

able to compose the rotation with another transformation (a reflection) that does not affect the octagon, and thereby achieve a symmetry of the extension, we say that the extension does in fact respect this symmetry of the octagon.

By comparing the first diagram with the last diagram we can check that the transformation (rotation followed by reflection) commutes with the projection operation, since from the first diagram it does not matter whether we project first and then rotate, or rotate (and reflect) and then project.

We can use the technique just described, of applying an optional reflection, to show that the extension in fact respects all symmetries of the octagon and is thus a fully symmetric extension of the octagon. We generalize this idea as follows: an extension is symmetric if, for every transformation of the feasible region that is a symmetry of the feasible region, there is some transformation (not necessarily the same one) that we can apply to the extension that also leaves the extension invariant, and keeps the extension in sync with its projection, in the sense that the transformation commutes with the projection operation.

When Yannakakis examined Swart's construction for the TSP described earlier, he observed that the proposed LP, if correct, would be a polynomial size symmetric extended formulation for the TSP. Yannakakis then showed that any symmetric extended formulation for the TSP must have exponential size, thus proving that the construction was incorrect. In the course of proving this, he also showed that any linear symmetric extended formulation for the matching problem must have exponential size.

Requiring that an extended formulation be symmetric is a restriction. In some cases the smallest asymmetric formulation for a symmetric problem is much smaller than the smallest symmetric formulation; refer to Section 1.8 and Chapter 7 for more discussion of this phenomenon. Given this fact, one may wonder why anyone would require symmetry or study symmetric formulations. We give several reasons below.

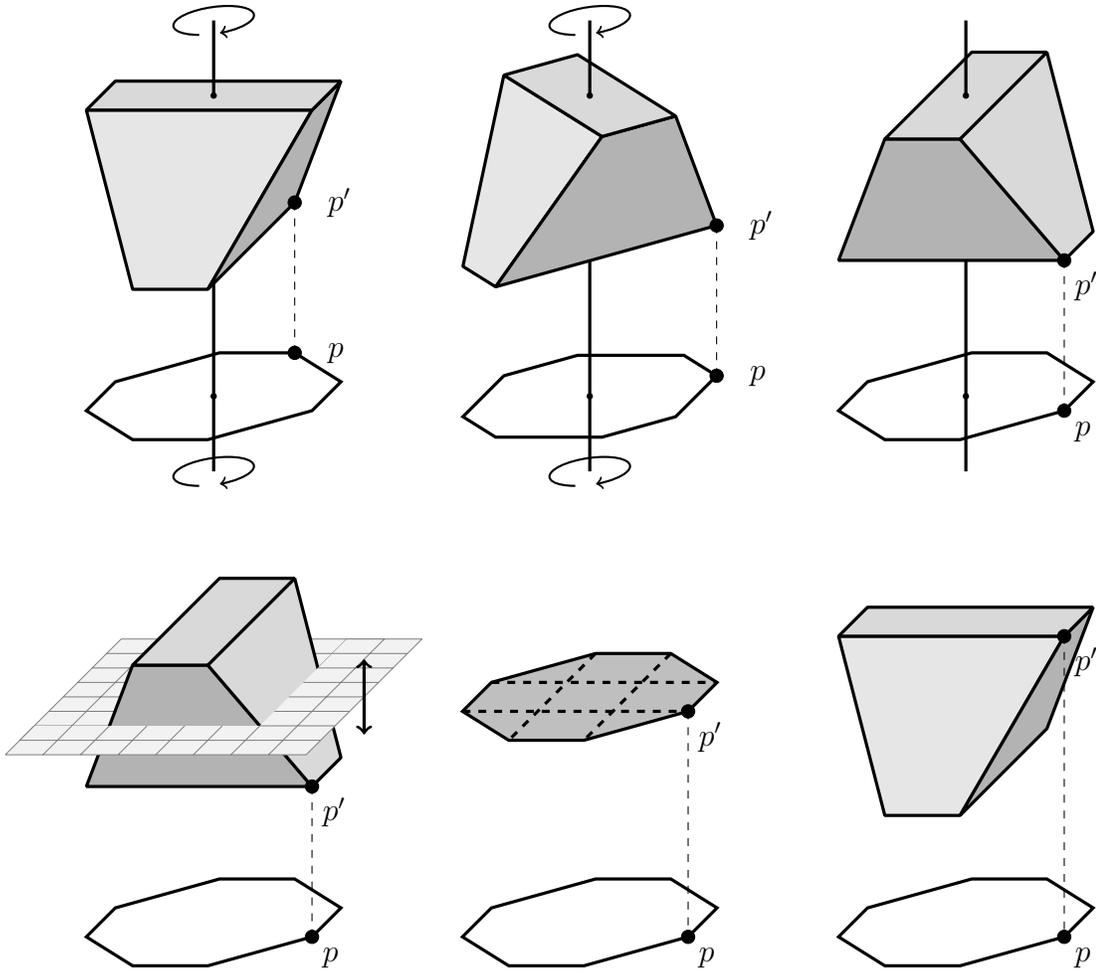


Figure 1.6: 90° rotational symmetry of the fruit basket problem. Top row: 90° rotation about the axis of symmetry. Bottom row: reflection through a plane parallel to the octagon.

1. As implied by Swart's example, symmetric formulations come about naturally when one is trying to solve a symmetric problem. Many candidate formulations for a symmetric problem are likely to be symmetric.
2. Symmetry is often preserved when one uses certain explicit constructions of extended formulations known as *hierarchies*. Refer to Section 1.8 for more discussion of hierarchies; also see Laurent [2003] for a survey.
3. A lower bound on symmetric formulations can rule out a wide range of approaches and guide algorithm designers in search of a small formulation to

focus on asymmetric cases only.

4. The symmetric case is often easier to reason about and can give insight into the asymmetric case. As a prime example, the insights that Yannakakis's developed in proving symmetric lower bounds were crucial to proving the asymmetric lower bounds that were derived later.
5. For certain classes of problems it is possible to prove that the best asymmetric formulation is not much smaller than the best symmetric formulation. In these cases any previously obtained symmetric lower bounds carry over to the asymmetric case.
6. Finally, it is sometimes the case that optimization algorithms can take advantage of symmetries in a formulation to perform better; see Dobre and Vera [2015] for an example.

1.6 Extended formulations as a model of computation

Understanding models of computation has been a core part of theoretical computer science since its inception. The Turing machine was the first general model of computation that was both mathematically adequate and intuitively compelling. The Turing machine lies behind the Church-Turing Thesis and indeed the P vs NP problem. Turing machines are in some ways extremely simple to reason about and in other ways extremely hard to prove anything about. This difficulty has prompted researchers to consider alternative models of computation as a way of gaining insight into problems such as P vs NP. One example is the study of circuit complexity, which blossomed in the 1980's. Polynomial size circuits are closely related to polynomial time Turing machines, yet have a combinatorial structure that permits deeper mathematical analysis.

Extended formulations can also be viewed as an alternate model of computation, and like circuits can potentially give insight into problems such as P vs NP. Linear and semidefinite extended formulations are related to polynomial time computable functions in the sense that the solution to a linear or semidefinite program with polynomial encoding length can be computed in polynomial time to any fixed accuracy.

Nonetheless, linear extended formulations and polynomial time computation are incomparable. On one hand, Edmonds [1965] showed that matchings can be computed in polynomial time while Rothvoß [2014] showed that the matching problem has no small LP formulation. On the other hand, it is an easy consequence of Balas [1998] that there are languages that are uncomputable (and therefore certainly not solvable in polynomial time) that have small LP formulations. This latter case arises because in the framework of extended formulations we have the freedom to construct different extended formulations for different sizes of the same problem. Formally, we would say that extended formulations are a *nonuniform* model of computation whereas algorithms are a *uniform* model.

The situation with semidefinite extended formulations is still open. While semidefinite formulations are also nonuniform and therefore include uncomputable functions, it is not known whether every polynomial time computable function has a small semidefinite program.

The situation with copositive extended formulations is also still open, but in a different sense. It is clear that any polynomial time computable function can be encoded as a small copositive program, but it is not clear what gap, if any, there is between functions computable by, say, small circuits and small copositive programs.

1.7 Contribution

We first show that there is no small symmetric SDP for the matching problem. This result first appeared in Braun et al. [2016] and is presented here with the kind permis-

sion of the Society for Industrial and Applied Mathematics (see Appendix A). Our result is an SDP analog of the result in Yannakakis [1991, 1988] that rules out a small symmetric LP for the matching problem. We note that our SDP lower bound also applies for approximating the matching problem. To prove our result we show that if the matching problem has a small symmetric SDP, then there is a low degree sum of squares refutation of the existence of a perfect matching in an odd clique, which contradicts a result by Grigoriev [2001].

We next define the notion of a symmetric conjunctive normal form (CNF) formula and show that any combinatorial problem that can be expressed by a small symmetric CNF has a small symmetric copositive formulation. We then give explicit constructions to show that both matching and TSP have small symmetric CNFs.

1.8 Related work

Some of the content of this Related Work section is adapted from Braun et al. [2016] and appears here with the kind permission of the Society for Industrial and Applied Mathematics (see Appendix A).

As mentioned previously, Yannakakis [1991, 1988] showed that any symmetric linear program for matching or TSP has exponential size. In doing so he began the systematic study of extended formulations. One of his key insights was that a linear extended formulation for a given problem corresponds to a nonnegative factorization of a combinatorial object associated with that problem, known as the *slack matrix*. In particular, the size of a minimal formulation is equal to the nonnegative rank of the slack matrix. Thus to find the smallest formulation it is not necessary to consider all possible higher dimensional extensions but instead it suffices to analyze a single quantity, namely the nonnegative rank of the slack matrix.

A natural question that came out of the work of Yannakakis is whether asymmetric formulations are more powerful than symmetric ones for symmetric problems.

Kaibel et al. [2010] showed that the problem of detecting matchings with a logarithmic number of edges in the complete graph has a polynomial size asymmetric linear formulation whereas any symmetric formulation has superpolynomial size. Goemans [2015] and Pashkovich [2014] showed that the smallest symmetric formulation for the permutahedron has quadratic size while the smallest asymmetric formulation is subquadratic.

Nonetheless, despite the evidence that asymmetry can sometimes add power, Fiorini et al. [2012, 2015b] showed that allowing asymmetry does not substantially improve the size of linear extended formulations for the TSP, and Rothvoß [2014] did the same for matching. In particular any linear extended formulation for either problem, symmetric or not, has exponential size.

Subsequently, Braun et al. [2012, 2015a], Chan et al. [2013], Braverman and Moitra [2013], Bazzi et al. [2015] generalized the framework of Yannakakis to give lower bounds for linear formulations that approximate combinatorial optimization problems. Braun et al. [2015c] and Braun et al. [2015b] generalized the reduction mechanism of extended formulations to abstract away the dependence on the choice of encoding for feasible solutions, and also to allow reductions that preserve approximation factors, even for fractional optimization problems.

For the class of maximum constraint satisfaction problems (MaxCSPs), Chan et al. [2013] established a connection between lower bounds for general linear programs and lower bounds against an explicit linear program, namely that defined by the hierarchy of Sherali and Adams [1990]. Using that connection, Chan et al. [2013] showed that a constant number of rounds of Sherali-Adams yields essentially as good an approximation as any polynomial size relaxation of a MaxCSP. By appealing to lower bounds on Sherali-Adams relaxations of MaxCSPs in literature, they then gave super-polynomial lower bounds for MAX3SAT and other MaxCSPs.

Given the general LP lower bounds, it is natural to ask whether the situation is

different for SDP relaxations. Semidefinite programs generalize linear programs and can be solved efficiently both in theory and practice (see Vandenberghe and Boyd [1996]). SDPs are the basis of some of the best algorithms currently known, for example the approximation of Goemans and Williamson [1995] for MAXCUT.

Following prior work (see for example Gouveia et al. [2011]) we define the size of an SDP formulation as the dimension of the psd cone from which the polytope can be obtained as an affine slice. This generalizes the nonnegative factorizations of Yannakakis to psd factorizations. Some recent work has shown limits to the power of small SDPs. Briët et al. [2013, 2015] nonconstructively give an exponential lower bound on the size of SDP formulations for most 0/1 polytopes.

Building on the approach of Chan et al. [2013], Lee et al. [2014] showed that for the class of MaxCSPs, the Lasserre SDP relaxation essentially yields the optimal symmetric SDP approximation. In light of known lower bounds for Lasserre SDP relaxations of MAX3SAT, this yields a corresponding lower bound for approximating MAX3SAT. In a significant recent advance, Lee et al. [2015] show an exponential lower bound even for *asymmetric* SDP relaxations of the TSP.

The state of lower bounds for matching and TSP are summarized in Table 1.1 on page 22.

Table 1.1: Exponential lower bounds for formulations of matching and TSP.

		Matching	TSP
LP	symmetric	Yannakakis [1991, 1988]	Yannakakis [1991, 1988]
	asymmetric	Rothvoß [2014]	Fiorini et al. [2015b, 2012]
SDP	symmetric	<i>this work</i>	see note ¹
	asymmetric	(open)	Lee et al. [2015]

¹ Lee et al. [2014] and Fawzi et al. [2015] give symmetric SDP formulation lower bounds for MaxCSPs and the cut polytope, respectively.

Turning now to copositive formulations, Maksimenko [2012] showed that any language definable as the set of solutions of a polynomial size CNF is a face of the cut polytope. Fiorini et al. [2015a] showed that the cut polytope has a small copositive formulation. They also define the *polynomially definable* languages and use the result of Maksimenko to show that every such language has a small copositive extension. We will show later that the class of polynomially definable languages is in fact a complexity class known as NP/poly, which is the class of languages computable in nondeterministic polynomial time with polynomial advice.

CHAPTER 2

MATHEMATICAL BACKGROUND

In this chapter we establish mathematical background that will be used in the rest of this document.

2.1 Notation

Let \mathbb{R} denote the real number line and let \mathbb{R}^d denote the standard d -dimensional Euclidean space with Cartesian coordinates. Elements of \mathbb{R}^d will be treated interchangeably as points and column vectors. Let $(\mathbb{R}^d)^*$ denote the dual space of \mathbb{R}^d . Thus $(\mathbb{R}^d)^*$ is the set of d -dimensional row vectors, or equivalently the set of linear functions from \mathbb{R}^d to \mathbb{R} .

The expression $[k]$ denotes the set of natural numbers $\{1, \dots, k\}$. The symbols \mathbb{R}_+ and \mathbb{R}_{++} denote the sets of nonnegative and strictly positive reals, respectively. The symbol e_i denotes the i th basis vector as a column vector, with dimension taken from context.

If M is a matrix then M_i denotes the i th row of M and M_{ij} denotes $(M_i)_j$, the (i, j) entry of M . Unless otherwise stated, if a function $f: \mathbb{R} \rightarrow \mathbb{R}$ is applied to a vector, matrix, or set, the function is assumed to act elementwise. Likewise, relational operators act elementwise unless otherwise stated. For example, if a and b are vectors of the same dimension, $a \geq b$ indicates that each element of a is greater than or equal to the corresponding element of b . In particular, $a \geq 0$ means that a is elementwise nonnegative.

The expression $\mathbb{R}^{n \times n}$ denotes the space of real $n \times n$ matrices, and the expression \mathbb{S}^n denotes the set of real $n \times n$ symmetric matrices. Note that \mathbb{S}^n is a subspace of $\mathbb{R}^{n \times n}$.

and that $\mathbb{R}^{n \times n}$ is isomorphic to \mathbb{R}^{n^2} . Let \mathbb{S}_+^r denote the cone of $r \times r$ real symmetric positive semidefinite (psd) matrices. Let $\mathbb{R}[x]$ denote the set of polynomials in n real variables $x = (x_1, \dots, x_n)$ with real coefficients. For a set $\mathcal{H} \subseteq \mathbb{R}[x]$ let $\langle \mathcal{H} \rangle$ denote the vector space spanned by \mathcal{H} and let $\langle \mathcal{H} \rangle_I$ denote the ideal generated by \mathcal{H} . The notation $\deg p$ denotes the degree of the polynomial p . If a group G acts on a set X , the (left) action of $g \in G$ on $x \in X$ is denoted $g \cdot x$.

The inner product of two vectors a and b of the same dimension is given by $a^\top b$ and the (Frobenius) inner product of two symmetric matrices A and B of the same dimension is given by $\text{Tr}[AB]$.

For sets $A, B \subseteq \mathbb{R}^d$, the notation $A + B$ denotes the *Minkowski sum*:

$$A + B := \{a + b \in \mathbb{R}^d \mid a \in A, b \in B\} .$$

If $x \in \mathbb{R}^d$ is a point then $x + B$ is shorthand for $\{x\} + B$.

The symbol S_n denotes the symmetric group on n letters. An element σ of S_n is a 1-1 and onto function from $[n]$ to $[n]$. Unless otherwise specified, the action of S_n on \mathbb{R}^n is permutation of coordinates, and the action of S_n on \mathbb{S}^n and $\mathbb{R}^{n \times n}$ is simultaneous permutation of rows and columns. We formalize this notion in the following definition.

Definition 1 (standard action). The *standard action* of S_n on \mathbb{R}^n is given by

$$\sigma \cdot (x_1, \dots, x_n) := (x_{\sigma^{-1}(1)}, \dots, x_{\sigma^{-1}(n)})$$

where $\sigma : [n] \rightarrow [n]$ is any element of S_n and $x = (x_1, \dots, x_n)$ is any element of \mathbb{R}^n . Similarly, the standard action of S_n on $\mathbb{R}^{n \times n}$ is given by

$$(\sigma \cdot X)_{i,j} := X_{\sigma^{-1}(i), \sigma^{-1}(j)}$$

where σ is as before and X is any element of $\mathbb{R}^{n \times n}$.

2.2 Basic definitions from convex geometry

We will make use of the following standard definitions from convex geometry.

Definition 2 (linear combination). A *linear combination* of the points

$$x_1, \dots, x_n$$

in \mathbb{R}^d is a point of the form

$$\sum_{i=1}^n \lambda_i x_i$$

with $\lambda_1, \dots, \lambda_n \in \mathbb{R}$.

By convention, an empty sum ($n = 0$) is allowed and is equal to $0 \in \mathbb{R}^d$.

Definition 3 (linear independence). A set of points is *linearly independent* if no point in the set can be expressed as a linear combination of the remaining points.

Definition 4 (linear hull (span)). The *linear hull* or *span* of a subset $X \subseteq \mathbb{R}^d$, denoted $\text{span } X$, is the set of all linear combinations of points in X .

The span of any set of points in \mathbb{R}^d is a *linear space*.

Definition 5 (linear dimension). The dimension of a linear space L , denoted $\dim X$, is the size of any basis for L .

Definition 6 (linear transformation (map)). A *linear transformation* or *linear map* from \mathbb{R}^n to \mathbb{R}^m is any function $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ that can be expressed as $f(x) = Ax$ for some real $m \times n$ matrix A .

Definition 7 (affine combination). An *affine combination* of the points x_1, \dots, x_n in \mathbb{R}^d is a point of the form

$$\sum_{i=1}^n \lambda_i x_i$$

with $\lambda_1, \dots, \lambda_n \in \mathbb{R}^d$ and $\sum_i \lambda_i = 1$.

Definition 8 (affine independence). A set of points is *affinely independent* if no point in the set can be expressed as an affine combination of the remaining points.

Definition 9 (affine hull). The *affine hull* of a set $X \subseteq \mathbb{R}^d$, denoted $\text{aff } X$, is the set of all affine combinations of points in X .

The affine hull of any nonempty set of points is an *affine space*.

Remark 10. Any affine space A can be expressed as $x + L$ where $x \in A$ and L is a linear space. The subspace L is uniquely determined whereas x is not (unless A consists of a single point).

It follows that any affine space can be regarded as a translation of a linear space.

Definition 11 (affine dimension). The dimension of an affine space A , denoted $\dim A$, is the dimension of the corresponding linear space L as in the previous remark.

Definition 12 (affine transformation (map)). An *affine transformation* or *affine map* from \mathbb{R}^n to \mathbb{R}^m is any function $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ that can be expressed as $f(x) = Ax + b$ for some real $m \times n$ matrix A and real vector $b \in \mathbb{R}^m$.

An affine transformation is a linear transformation followed by a translation. It follows that every affine space is the image of a linear space under an affine transformation, and vice versa.

Remark 13. An injective affine transformation can be regarded as an (affine) change of coordinates. It will turn out that many of the properties we are interested in (for example, the extension complexity of a polytope, to be defined later) are preserved under affine changes of coordinates.

Definition 14 (conic combination). A *conic combination* or *nonnegative combination*

of the points x_1, \dots, x_n in \mathbb{R}^d is a point of the form

$$\sum_{i=1}^n \lambda_i x_i$$

with $\lambda_1, \dots, \lambda_n \in \mathbb{R}^d$ and each $\lambda_i \geq 0$.

Definition 15 (conic hull). The *conic hull* of a set $X \subseteq \mathbb{R}^d$, denoted $\text{cone } X$, is the set of all conic combinations of points in X .

Definition 16 (convex combination). A *convex combination* of the points x_1, \dots, x_n in \mathbb{R}^d is a point of the form

$$\sum_{i=1}^n \lambda_i x_i$$

with $\lambda_1, \dots, \lambda_n \in \mathbb{R}^d$, $\sum_i \lambda_i = 1$, and each $\lambda_i \geq 0$.

Definition 17 (convex hull). The *convex hull* of a set $X \subseteq \mathbb{R}^d$, denoted $\text{conv } X$, is the set of all convex combinations of points in X .

Definition 18 (hyperplane). Let $a \in \mathbb{R}^d$ be nonzero and let $b \in \mathbb{R}$. The $(d - 1)$ -dimensional affine space given by

$$\{x \in \mathbb{R}^d \mid a^\top x = b\}$$

is the *hyperplane* in \mathbb{R}^d defined by the equation $a^\top x = b$.

Definition 19 (halfspace). Let $a \in \mathbb{R}^d$ be nonzero and let $b \in \mathbb{R}$. The set

$$\{x \in \mathbb{R}^d \mid a^\top x \leq b\}$$

is the (closed) *halfspace* in \mathbb{R}^d defined by the inequality $a^\top x \leq b$.

2.3 Polytopes

In this section we present necessary background on polytopes. A standard reference for this topic is Ziegler [1995]; see also Brøndsted [1983]. Propositions given in this section without proof are proven in one of these references.

Definition 20 (polytope). A *polytope* in \mathbb{R}^d is any set of the form $\text{conv}(V)$ where V is a finite subset of \mathbb{R}^d .

Definition 21 (polyhedron). A *polyhedron* in \mathbb{R}^d is the (possibly empty) intersection of a finite number of closed halfspaces in \mathbb{R}^d .

It follows that a polyhedron in \mathbb{R}^d can be expressed as the set

$$\{x \in \mathbb{R}^d \mid Ax \leq b\}$$

for some $A \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$, where m is the number of halfspaces in the intersection. Informally we will refer interchangeably to $Ax \leq b$ as a linear system of inequalities and as the polyhedron $\{x \mid Ax \leq b\}$.

The following nontrivial fact is well-known; see Ziegler [1995] for a proof.

Proposition 22. *A subset of \mathbb{R}^d is a polytope (as defined above) iff it is a bounded polyhedron.*

The presentation of a polytope in the form $\text{conv } V$ is called an *inner description*, while the presentation of a polyhedron in the form $\{x \mid Ax \leq b\}$ is called an *outer description*.

Definition 23 (dimension of a polyhedron). The dimension of a polyhedron P is denoted $\dim P$ and is defined to be equal to $\dim(\text{aff } P)$.

Definition 24 (valid inequality). An inequality $cx \leq \delta$ is said to be *valid* for a polyhedron $P \subset \mathbb{R}^d$ if the inequality is satisfied by every point in P .

We will say that the equality $cx = \delta$ is a *valid hyperplane* for P if either $cx \leq \delta$ or $cx \geq \delta$ is valid for P .

Definition 25 (face). Let $P \subset \mathbb{R}^d$ be a polyhedron. A set $F \subset \mathbb{R}^d$ is a *face* of P iff there is an inequality $cx \leq \delta$ that is valid for P such that $F = \{x \mid x \in P \text{ and } cx = \delta\}$. In other words, every face of P is the intersection of P with a valid hyperplane.

Proposition 26. *Every face of a polyhedron is again a polyhedron.*

Note that by definition, both P itself and the empty set are faces of P : use the inequalities $0x \leq 0$ and $0x \leq 1$, respectively.

Definition 27 (proper face). If F is a face of the polyhedron P and F is not equal to P then F is a *proper face* of P .

Definition 28 (facet). A *facet* of a polyhedron P is a face of dimension $\dim P - 1$.

Definition 29 (vertex). A *vertex* of a polyhedron is a face of dimension 0 (that is, a point).

The set of all vertices of a polyhedron P is denoted $\text{vert } P$.

Definition 30 (edge). An *edge* of a polytope is a face of dimension 1 (that is, a line segment).

The following definition is slightly informal but hopefully clear. Refer to Ziegler [1995] for a formal definition.

Definition 31 (relative interior). Let $P \subseteq \mathbb{R}^d$ be a polyhedron. A point x is in the *relative interior* of P , denoted $\text{relint } P$, if x is in the interior of P when P is embedded in $\text{aff } P$ (in which P is full-dimensional).

Proposition 32. *If P is a polytope then $P = \text{conv}(\text{vert } P)$.*

2.3.1 Projection

Proposition 33. *The image of a polytope under an affine map is a polytope.*

Definition 34 (affinely isomorphic). The polytopes $P \subset \mathbb{R}^d$ and $Q \subset \mathbb{R}^e$ are *affinely isomorphic* if there is an affine map $f : \mathbb{R}^d \rightarrow \mathbb{R}^e$ such that $f(P) = Q$ and f is injective on P .

Definition 35 (projection). Let $\pi : \mathbb{R}^d \rightarrow \mathbb{R}^e$ be an affine map and let $P \subset \mathbb{R}^d$ be a polytope. Then $\pi(P)$ is the *projection* of P under π .

Informally, we will use the term projection interchangeably to refer both to the projection map π and the image $\pi(P)$ of P under the projection. As implied by the name, the projection $\pi(P)$ will typically be a lower dimensional polytope than P is.

2.4 Slack Matrices

Let $P \subset \mathbb{R}^d$ be a polytope with an associated inner and outer description:

$$P = \text{conv } V = \{x \mid Ax \leq b\},$$

where $V = \{v_1, \dots, v_n\} \subset \mathbb{R}^d$ is finite, A is an $m \times d$ real matrix, and $b \in \mathbb{R}^m$ is a vector. The following object will be central to our study.

Definition 36 (slack matrix). Let P be a polytope as above. The *slack matrix* of P (with respect to V , A , and b) is the $m \times n$ matrix S whose ij th entry is

$$S_{ij} := b_i - A_i v_j,$$

the slack of the j th element of V with respect to the i th inequality.

Note that the slack matrix is always nonnegative, corresponding to the fact that $V \subseteq P$.

Remark 37. We can define slack matrices more generally, with respect to any polyhedron $P := \{x \mid Ax \leq b\}$ and finite set $V \subset P$.

Definition 38 (correlation polytope). The *correlation polytope* $\text{COR}(n)$ is the convex hull of all rank-1 symmetric $n \times n$ 0/1 matrices:

$$\text{COR}(n) := \text{conv} \{bb^\top \in \mathbb{R}^{n \times n} \mid b \in \{0, 1\}^n\}.$$

In order to define our next object we need the notion of a cut. Let K_n denote the complete graph with vertex set $[n]$. For any subset $X \subseteq [n]$ let the *cut* defined by X be the set of edges with exactly one endpoint in X , and let $\delta(X) \in \mathbb{R}^{\binom{n}{2}}$ denote the characteristic vector of the cut defined by X :

$$\delta(X)_{ij} = \begin{cases} 1 & |X \cap \{i, j\}| = 1 \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i < j \leq n$.

Definition 39 (cut polytope). The *cut polytope* is the convex hull of all cut vectors of the complete graph:

$$\text{CUT}(n) := \text{conv} \left\{ \delta(X) \in \mathbb{R}^{\binom{n}{2}} \mid X \subseteq [n] \right\}.$$

We will later make use of the following well-known fact.

Theorem 40 ([De Simone, 1989/90]). *For all n , $\text{COR}(n)$ is linearly isomorphic to $\text{CUT}(1+n)$.*

Definition 41 (G -symmetric). Let G be a group acting on a Euclidean space E and let $S \subseteq E$ be a set. The set S is G -*symmetric* or G -*invariant* if the action of G on E leaves S unchanged:

$$g \cdot S = S$$

for all $g \in G$, where $g \cdot S$ is defined as $\{g \cdot x \mid x \in S\}$.

For our purposes the Euclidean space E in Definition 41 will usually be either \mathbb{R}^n or $\mathbb{R}^{n \times n}$.

Observation 42. The polytope $\text{COR}(n)$ is S_n -symmetric, assuming the standard action of S_n on $\mathbb{R}^{n \times n}$ as in Definition 1.

Definition 43 (copositive). A matrix $M \in \mathbb{S}^n$ is *copositive* if

$$x^\top M x \geq 0$$

whenever $x \geq 0$. The symbol \mathcal{C}_n denotes the set of $n \times n$ copositive matrices.

Definition 44 (completely positive). A matrix $M \in \mathbb{S}^n$ is *completely positive* if

$$M = BB^\top$$

for some entrywise nonnegative matrix B . The symbol \mathcal{C}_n^* denotes the set of $n \times n$ completely positive matrices.

Remark 45. The sets \mathcal{C}_n and \mathcal{C}_n^* form closed convex cones in \mathbb{S}^n that are dual to each other, as implied by the notation.

Remark 46. Both \mathcal{C}_n and \mathcal{C}_n^* are S_n -symmetric under the standard action given in Definition 1.

Definition 47 (extension). Let $S \in \mathbb{R}^n$ be a set, let $C \in \mathbb{R}^d$ be a closed convex cone, let $L \in \mathbb{R}^d$ be an affine space, and let $\pi : \mathbb{R}^d \rightarrow \mathbb{R}^n$ be an affine map. If

$$S = \pi(L \cap C)$$

then the set $L \cap C$ in \mathbb{R}^d is a C -extension of S via the map π .

Definition 48 (symmetric extension). Let S , C , L , and π be as in Definition 47, with $S = \pi(L \cap C)$. Let G be a group acting on \mathbb{R}^n and \mathbb{R}^d . If S , L and C are all G -symmetric, and

$$g \cdot \pi(x) = \pi(g \cdot x)$$

for all $x \in L \cap C$ and $g \in G$, then the set $L \cap C$ is a G -symmetric extension of S .

Definition 49 (extension complexity). Let \mathcal{C} be a family of closed convex cones parametrized by d , so that \mathcal{C} has the form

$$\mathcal{C} = \{C_d\}_{d \in \mathbb{N}}.$$

Let $S \in \mathbb{R}^n$ be a set. The smallest d , if any, such that S has a C_d -extension is the *extension complexity* of S with respect to the family \mathcal{C} .

Definition 50 (polynomial extension complexity). Let

$$\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$$

be a family of sets and let \mathcal{C} be a family of closed convex cones parametrized by d . If there is a polynomial p such that for each n the \mathcal{C} extension complexity of P_n is at most $p(n)$, then the family \mathcal{P} has *polynomial extension complexity* with respect to \mathcal{C} .

Definition 51 (CNF). A *CNF* is a Boolean formula in conjunctive normal form – that is, a Boolean formula consisting of ANDs of clauses, where each clause is an OR of literals, and each literal is a Boolean variable or its negation.

2.5 Linear, semidefinite, and copositive programming

A *linear program* is an optimization problem of the form

$$\text{minimize } c^\top x \text{ subject to } Ax = b \text{ and } x \geq 0,$$

where c and x are vectors in \mathbb{R}^d for some d , b is a vector in \mathbb{R}^m for some m , and A is a $d \times m$ real matrix. A , b , and c are given, and the problem is to find the minimum value of the objective function $c^\top x$ over all feasible vectors x . Note that the feasible region is the intersection of the hyperplane $Ax = b$ with the nonnegative cone \mathbb{R}_+^d .

A *semidefinite program* is an optimization problem of the form

$$\text{minimize } \text{Tr}[CX] \text{ subject to } \mathcal{A}(X) = b \text{ and } X \in \mathbb{S}_+^d,$$

where C and X are matrices in \mathbb{S}^d for some d , b is a vector in \mathbb{R}^m for some m , and \mathcal{A} is an affine linear operator from \mathbb{S}^d to \mathbb{R}^m . \mathcal{A} , b , and C are given, and the problem is to find the minimum value of the objective function $\text{Tr}[CX]$ over all feasible matrices X . Note that the feasible region is the intersection of the hyperplane $Ax = b$ with the semidefinite cone \mathbb{S}_+^d .

A *copositive program* is an optimization problem of the form

$$\text{minimize } \text{Tr}[CX] \text{ subject to } \mathcal{A}(X) = b \text{ and } X \in \mathcal{C}_d,$$

where C and X are matrices in \mathbb{S}^d for some d , b is a vector in \mathbb{R}^m for some m , and \mathcal{A} is an affine linear operator from \mathbb{S}^d to \mathbb{R}^m . \mathcal{A} , b , and C are given, and the problem is to find the minimum value of the objective function $\text{Tr}[CX]$ over all feasible matrices X . Note that the feasible region is the intersection of the hyperplane $Ax = b$ with the copositive cone \mathcal{C}_d .

CHAPTER 3

SYMMETRIC SDP EXTENDED FORMULATIONS

The contents of this chapter and Chapter 4 first appeared in an abridged form in Braun et al. [2016] and are reproduced here with the kind permission of the Society for Industrial and Applied Mathematics (see Appendix A).

3.1 Symmetric SDP formulations

In this section we define a framework for symmetric semidefinite programming formulations and show that a symmetric SDP formulation implies a symmetric sum of squares representation over a small basis. Our framework extends the one in Braun et al. [2015c] with a symmetry condition; see also Lee et al. [2014].

We now present our SDP formulation framework. We restrict ourselves to maximization problems even though the framework extends to minimization problems. A *maximization problem* $\mathcal{P} = (\mathcal{S}, \mathcal{F})$ consists of a finite set \mathcal{S} of feasible solutions and a finite set \mathcal{F} of nonnegative objective functions. Given two functions $\tilde{C}, \tilde{S}: \mathcal{F} \rightarrow \mathbb{R}$ specifying approximation guarantees, an algorithm (\tilde{C}, \tilde{S}) -approximately solves \mathcal{P} if for all $f \in \mathcal{F}$ with $\max_{s \in \mathcal{S}} f(s) \leq \tilde{S}(f)$ it computes $\tilde{f} \in \mathbb{R}$ satisfying $\max_{s \in \mathcal{S}} f(s) \leq \tilde{f} \leq \tilde{C}(f)$.

Remark 52. For an exact extension of a polytope

$$P = \text{conv}(V) = \{x \mid a_j x \leq b_j, j \in [m]\}$$

using this framework, we would define $f_j(x) := b_j - a_j x$ for each $j \in [m]$ and then set $\mathcal{S} = V$, $\mathcal{F} = \{f_j \mid j \in [m]\}$, and $\tilde{C}(f) = \tilde{S}(f) = \max_{x \in P} f(x)$ for all $f \in \mathcal{F}$.

Let G be a group with associated actions on \mathcal{S} and \mathcal{F} . The problem \mathcal{P} is G -*symmetric* if the group action satisfies the compatibility constraint $(g \cdot f)(g \cdot s) = f(s)$. For a G -symmetric problem we require G -symmetric approximation guarantees: $\tilde{C}(g \cdot f) = \tilde{C}(f)$ and $\tilde{S}(g \cdot f) = \tilde{S}(f)$ for all $f \in \mathcal{F}$ and $g \in G$.

We now define the notion of a semidefinite programming formulation of a maximization problem.

Definition 53 (SDP formulation for \mathcal{P}). Let $\mathcal{P} = (\mathcal{S}, \mathcal{F})$ be a maximization problem with approximation guarantees \tilde{C}, \tilde{S} . A (\tilde{C}, \tilde{S}) -*approximate SDP formulation of \mathcal{P}* of size d consists of a linear map $\mathcal{A}: \mathbb{S}_+^d \rightarrow \mathbb{R}^k$ and $b \in \mathbb{R}^k$ together with:

1. *Feasible solutions*: an $X^s \in \mathbb{S}_+^d$ with $\mathcal{A}(X^s) = b$ for all $s \in \mathcal{S}$, i.e., the SDP $\{X \in \mathbb{S}_+^d \mid \mathcal{A}(X) = b\}$ is a relaxation of $\text{conv} \{X^s \mid s \in \mathcal{S}\}$,
2. *Objective functions*: an affine function $w^f: \mathbb{S}_+^d \rightarrow \mathbb{R}$ satisfying

$$w^f(X^s) = f(s)$$

for all $f \in \mathcal{F}$ with $\max_{s \in \mathcal{S}} f(s) \leq \tilde{S}(f)$ and all $s \in \mathcal{S}$, i.e., the linearizations are exact on solutions, and

3. *Performance guarantee*: $\max \{w^f(X) \mid \mathcal{A}(X) = b, X \in \mathbb{S}_+^d\} \leq \tilde{C}(f)$ for all $f \in \mathcal{F}$ with $\max_{s \in \mathcal{S}} f(s) \leq \tilde{S}(f)$.

If G is a group, \mathcal{P} is G -symmetric, and G acts on \mathbb{S}_+^d , then an SDP formulation of \mathcal{P} with symmetric approximation guarantees \tilde{C}, \tilde{S} is G -*symmetric* if it additionally satisfies the compatibility conditions for all $g \in G$:

1. *Action on solutions*: $X^{g \cdot s} = g \cdot X^s$ for all $s \in \mathcal{S}$,
2. *Action on functions*:

$$w^{g \cdot f}(g \cdot X) = w^f(X)$$

for all $f \in \mathcal{F}$ with $\max_{s \in \mathcal{S}} f(s) \leq \tilde{S}(f)$, and

3. *Invariant affine space:* $\mathcal{A}(g \cdot X) = \mathcal{A}(X)$.

A G -symmetric SDP formulation is G -coordinate-symmetric if the action of G on \mathbb{S}_+^d is by permutation of coordinates: that is, there is an action of G on $[d]$ with $(g \cdot X)_{ij} = X_{g^{-1} \cdot i, g^{-1} \cdot j}$ for all $X \in \mathbb{S}_+^d$, $i, j \in [d]$ and $g \in G$.

3.2 The symmetric factorization lemma

In this section we turn a G -coordinate-symmetric SDP formulation into a symmetric sum of squares representation over a small set of basis functions.

We first develop some facts that will be used in the proof of the factorization lemma. Recall that for a matrix $M \in \mathbb{S}_+^d$, \sqrt{M} denotes the unique psd matrix such that $\sqrt{M}\sqrt{M} = M$.

Fact 54. *Let G be a group that acts on \mathbb{S}_+^d by simultaneous permutation of rows and columns. Then $\sqrt{g \cdot X} = g \cdot \sqrt{X}$ for any $g \in G$ and $X \in \mathbb{S}_+^d$.*

Proof. By the assumed action of G on \mathbb{S}_+^d , for any $g \in G$ there is a permutation matrix $\phi(g) \in \mathbb{R}^{d \times d}$ with $g \cdot X = \phi(g)X\phi(g)^\top$ for any $X \in \mathbb{S}_+^d$. Using the orthogonality of permutation matrices we find that

$$\begin{aligned}
(g \cdot \sqrt{X})^2 &= (\phi(g)\sqrt{X}\phi(g)^\top)^2 \\
&= \phi(g)\sqrt{X}\phi(g)^\top\phi(g)\sqrt{X}\phi(g)^\top \\
&= \phi(g)\sqrt{X}I\sqrt{X}\phi(g)^\top \\
&= \phi(g)X\phi(g)^\top \\
&= g \cdot X \\
&= (\sqrt{g \cdot X})^2.
\end{aligned}$$

Since $g \cdot \sqrt{X}$ and $\sqrt{g \cdot X}$ are both psd, we can take the (unique psd) square root of both sides to complete the proof. \square

Lemma 55. *Let*

$$\left(\mathcal{A}, b, \{X^s\}_{s \in \mathcal{S}}, \{w^f\}_{f \in \mathcal{F}} \right)$$

comprise a (\tilde{C}, \tilde{S}) -approximate SDP formulation of size d for the maximization problem $\mathcal{P} = (\mathcal{S}, \mathcal{F})$. Then for every $f \in \mathcal{F}$ with $\max f \leq \tilde{S}(f)$, there is a $U^f \in \mathbb{S}_+^d$ and a $\mu_f \geq 0$ such that for all $s \in \mathcal{S}$,

$$\tilde{C}(f) - f(s) = \text{Tr}[U^f X^s] + \mu_f.$$

Proof. We may assume the SDP is strictly feasible, since otherwise the spectrahedron $\{X \in \mathbb{S}_+^d \mid \mathcal{A}(X) = b\}$ is contained in a proper face of \mathbb{S}_+^d , which is a psd cone of strictly smaller size. Let $f \in \mathcal{F}$ be such that $\max f \leq \tilde{S}(f)$, let $w^f(X)$ be given by $\text{Tr}[CX] + c$, and consider the SDP

$$\max \left\{ \text{Tr}[CX] \mid \mathcal{A}(X) = b, X \in \mathbb{S}_+^d \right\}.$$

Let δ^* denote the value of this SDP. By assumption, $\delta^* + c \leq \tilde{C}(f)$; define $\mu_f = \tilde{C}(f) - \delta^* - c \geq 0$. Because the SDP is bounded and strictly feasible we can apply strong duality to conclude that it is equal to

$$\min \left\{ b^\top y \mid \sum_i y_i \mathcal{A}_i - C \in \mathbb{S}_+^d \right\}.$$

Let y^* be a solution of the dual program. Note that $\delta^* = b^\top y^* = \sum_i y_i^* b_i$. Define $U^f = \sum y_i^* \mathcal{A}_i - C$. Note that U^f is psd and that $C = \sum y_i^* \mathcal{A}_i - U^f$. For every $s \in \mathcal{S}$ we now have

$$\begin{aligned}
\tilde{C}(f) - f(s) &= \tilde{C}(f) - w^f(X^s) \\
&= \tilde{C}(f) - (\text{Tr}[CX^s] + c) \\
&= \tilde{C}(f) - \delta^* + \delta^* - \text{Tr} \left[\left(\sum_i y_i^* \mathcal{A}_i - U^f \right) X^s \right] - c \\
&= \tilde{C}(f) - \delta^* + \sum_i y_i^* b_i - \sum_i y_i^* \text{Tr}[\mathcal{A}_i X^s] + \text{Tr}[U^f X^s] - c \\
&= \sum_i y_i^* (b_i - \text{Tr}[\mathcal{A}_i X^s]) + \text{Tr}[U^f X^s] + (\tilde{C}(f) - \delta^* - c) \\
&= \text{Tr}[U^f X^s] + \mu_f,
\end{aligned}$$

where in the last step we have used the fact that $\mathcal{A}(X^s) = b$ for $s \in \mathcal{S}$. \square

Fact 56. If $A, B \in \mathbb{S}_+^d$ then $\text{Tr}[AB] = \sum_{i,j} \left(\sum_k \sqrt{A_{ik}} \sqrt{B_{kj}} \right)^2$. In particular, the trace of the product is a sum of squares.

Proof. Here we use the cyclic property of the trace, namely $\text{Tr}[ABC] = \text{Tr}[BCA]$, the fact that $M = M^\top$ when M is symmetric, and the fact that $\text{Tr}[M^\top M] = \sum_{i,j} (M_{ij})^2$ for any matrix M :

$$\begin{aligned}
\text{Tr}[AB] &= \text{Tr} \left[\sqrt{A} \sqrt{A} \sqrt{B} \sqrt{B} \right] \\
&= \text{Tr} \left[\sqrt{B} \sqrt{A} \sqrt{A} \sqrt{B} \right] \\
&= \text{Tr} \left[\left(\sqrt{A} \sqrt{B} \right)^\top \left(\sqrt{A} \sqrt{B} \right) \right] \\
&= \sum_{i,j} \left(\left(\sqrt{A} \sqrt{B} \right)_{ij} \right)^2 \\
&= \sum_{i,j} \left(\sum_k \sqrt{A_{ik}} \sqrt{B_{kj}} \right)^2.
\end{aligned}$$

\square

We are now ready to prove the main lemma of this section.

Lemma 57 (Sum of squares for a symmetric SDP formulation). *If a G -symmetric maximization problem*

$$\mathcal{P} = (\mathcal{S}, \mathcal{F})$$

admits a G -coordinate-symmetric (\tilde{C}, \tilde{S}) -approximate SDP formulation of size d , then there is a set \mathcal{H} of at most $\binom{d+1}{2}$ functions $h: \mathcal{S} \rightarrow \mathbb{R}$ such that for any $f \in \mathcal{F}$ with $\max f \leq \tilde{S}(f)$ we have $\tilde{C}(f) - f = \sum_j h_j^2 + \mu_f$ for some $h_j \in \langle \mathcal{H} \rangle$ and constant $\mu_f \geq 0$. Furthermore the set \mathcal{H} is invariant under the action of G given by $(g \cdot h)(s) = h(g^{-1} \cdot s)$ for $g \in G$, $h \in \mathcal{H}$ and $s \in \mathcal{S}$.

Proof. Let

$$\left(\mathcal{A}, b, \{X^s\}_{s \in \mathcal{S}}, \{w^f\}_{f \in \mathcal{F}} \right)$$

comprise a G -coordinate-symmetric SDP formulation of size d . We define the set $\mathcal{H} := \{h_{ij} \mid i, j \in [d]\}$ via $h_{ij}(s) := \sqrt{X^s}_{ij}$. We first show that $g \cdot h_{ij} = h_{g \cdot i, g \cdot j}$ (in other words, \mathcal{H} is G -symmetric):

$$\begin{aligned} (g \cdot h_{ij})(s) &= h_{ij}(g^{-1} \cdot s) && \text{(action on } \mathcal{H} \text{)} \\ &= \left(\sqrt{X^{(g^{-1} \cdot s)}} \right)_{ij} && \text{(definition)} \\ &= \left(\sqrt{g^{-1} \cdot X^s} \right)_{ij} && \text{(action on solutions)} \\ &= \left(g^{-1} \cdot \sqrt{X^s} \right)_{ij} && \text{(Fact 54)} \\ &= \sqrt{X^s}_{g \cdot i, g \cdot j} && \text{(action on } \mathbb{S}_+^d \text{)} \\ &= h_{g \cdot i, g \cdot j}(s). && \text{(definition)} \end{aligned}$$

Since $h_{ij} = h_{ji}$, the set \mathcal{H} has at most $\binom{d+1}{2}$ elements. Pick any $f \in \mathcal{F}$ with

$\max f \leq \tilde{S}(f)$. Using Lemma 55 we have that there exist $U^f \in \mathbb{S}_+^d$ and $\mu_f \geq 0$ such that $\tilde{C}(f) - f(s) = \text{Tr}[U^f X^s] + \mu_f$ for all $s \in \mathcal{S}$. Finally, using Fact 56 we derive that

$$\tilde{C}(f) - f(s) = \sum_{i,j} \left(\sum_k \sqrt{U^f_{ik}} \sqrt{X^s_{kj}} \right)^2 + \mu_f,$$

and thus

$$\tilde{C}(f) - f = \sum_{i,j} \left(\sum_k \sqrt{U^f_{ik}} h_{kj} \right)^2 + \mu_f.$$

Since U^f is a constant matrix, for any i, j, k we have $\sum_k \sqrt{U^f_{ik}} h_{kj} \in \langle \mathcal{H} \rangle$, so in the last equation $\tilde{C}(f) - f$ is expressed in the form $\sum_j h_j^2 + \mu_f$ with each $h_j \in \langle \mathcal{H} \rangle$ and $\mu_f \geq 0$. □

CHAPTER 4

A SYMMETRIC SDP LOWER BOUND FOR MATCHING

The contents of this chapter and Chapter 3 first appeared in an abridged form in Braun et al. [2016] and are reproduced here with the kind permission of the Society for Industrial and Applied Mathematics (see Appendix A).

4.1 The perfect matching problem

We present the *perfect matching problem* $\text{PM}(n)$ as a maximization problem in the framework of Section 3.1 and show that any symmetric SDP formulation for it has exponential size.

Let n be an even positive integer, and let K_n denote the complete graph on n vertices. The feasible solutions of $\text{PM}(n)$ are all the perfect matchings M on K_n . The objective functions f_F are indexed by the edge sets F of K_n and are defined as $f_F(M) := |M \cap F|$. For approximation guarantees we use $\tilde{S}(f) := \max f$ and $\tilde{C}(f) := \max f + \varepsilon/2$ for some fixed $0 \leq \varepsilon < 1$ as in Braun and Pokutta [2015a]; see also Braun and Pokutta [2015b] for a more in-depth discussion.

Since $\tilde{S}(f) = \max f \leq (n-1)/2$ when f is associated with an odd set, we have $(1 - \varepsilon/(n-1))\tilde{C}(f) \geq \tilde{S}(f)$, which will establish an inapproximability ratio of $1 - \varepsilon/(n-1)$. Refer to Section 4.2 for a more detailed discussion of the ratio.

4.1.1 Symmetric functions on matchings are juntas

In this section we show that functions on perfect matchings with high symmetry are actually *juntas*: they depend only on the edges of a small vertex set. The key is the following lemma stating that perfect matchings coinciding on a vertex set belong to

the same orbit of the pointwise stabilizer of the vertex set. For any set $W \subseteq [n]$ let $E[W]$ denote the edges of K_n with both endpoints in W .

Lemma 58. *Let $S \subseteq [n]$ with $|S| < n/2$ and let M_1 and M_2 be perfect matchings in K_n . If $M_1 \cap E[S] = M_2 \cap E[S]$ then there exists $\sigma \in A([n] \setminus S)$ such that $\sigma \cdot M_1 = M_2$.*

Proof. Let $\delta(S)$ denote the edges with exactly one endpoint in S . There are three kinds of edges: those in $E[S]$, those in $\delta(S)$, and those disjoint from S . We construct σ to handle each type of edge, then fix σ to be even.

To handle the edges in $E[S]$ we set σ to the identity on S , since $M_1 \cap E[S] = M_2 \cap E[S]$.

To handle the edges in $\delta(S)$ we note that $V(M_1 \cap \delta(S))$ equals $V(M_2 \cap \delta(S))$ when both are restricted to S , since M_1 and M_2 are perfect matchings. Therefore for each edge $(s, v) \in M_1$ with $s \in S$ and $v \notin S$ there is a unique edge $(s, w) \in M_2$ with $w \notin S$; we extend σ to map v to w for each such s .

To handle the edges disjoint from S , we again use the fact that M_1 and M_2 are perfect matchings, so the number of edges in each that are disjoint from S is the same. We extend σ to be an arbitrary bijection on those edges.

We now show that we can choose σ to be even. Since $|S| < n/2$ there is an edge $(u, v) \in M_2$ disjoint from S . Let $\tau_{u,v}$ denote the transposition of u and v and let $\sigma' := \tau_{u,v} \circ \sigma$. We have $\sigma' \cdot M_1 = \sigma \cdot M_1 = M_2$, and either σ or σ' is even. \square

We also need the following lemma, which has been used extensively for symmetric linear extended formulations. See references Yannakakis [1988, 1991], Kaibel et al. [2010], Braun and Pokutta [2011], Lee et al. [2014] for examples.

Lemma 59 ([Dixon and Mortimer, 1996, Theorems 5.2A and 5.2B]). *Let $n \geq 10$ and let $G \leq A_n$ be a group. If $|A_n : G| < \binom{n}{k}$ for some $k < n/2$, then there is a subset $W \subseteq [n]$ such that $|W| < k$, W is G -invariant, and $A([n] \setminus W)$ is a subgroup of G .*

We now formally state and prove the claim about juntas:

Proposition 60. *Let $n \geq 10$, let $k < n/2$ and let \mathcal{H} be an A_n -symmetric set of functions on the set of perfect matchings of K_n of size less than $\binom{n}{k}$. Then for every $h \in \mathcal{H}$ there is a vertex set $W \subseteq [n]$ of size less than k such that h depends only on the (at most $\binom{k-1}{2}$) edges in W .*

Proof. Applying Lemma 59 to the stabilizer of h , we obtain a subset $W \subseteq [n]$ of size less than k such that h is stabilized by $A([n] \setminus W)$. In other words, we have

$$h(M) = (g \cdot h)(M) = h(g^{-1} \cdot M)$$

for all $g \in A([n] \setminus W)$.

Therefore for every perfect matching M the function h is constant on the orbit of M corresponding to $A([n] \setminus W)$. Lemma 58 shows that the orbit is determined by $M \cap E[W]$, from which it follows that the function value $h(M)$ is also. Therefore h depends only on the edges in $E[W]$. \square

4.1.2 The matching polynomials

A key step in proving our lower bound is obtaining low-degree derivations of approximation guarantees for objective functions of $\text{PM}(n)$. Therefore we start with a standard representation of functions as polynomials. We define the *matching constraint polynomials* as

$$\begin{aligned} \mathcal{P}_n := & \{x_{uv}x_{uw} \mid u, v, w \in [n] \text{ distinct}\} \\ & \cup \left\{ \sum_{u \in [n], u \neq v} x_{uv} - 1 \mid v \in [n] \right\} \\ & \cup \{x_{uv}^2 - x_{uv} \mid u, v \in [n] \text{ distinct}\} . \end{aligned} \tag{4.1}$$

Intuitively, the first set of polynomials ensures that no vertex is matched more than once, the second set ensures that each vertex is matched, and the third set ensures

that each coordinate is 0-1 valued. We observe that the ring of real valued functions on perfect matchings is isomorphic to

$$\mathbb{R}[\{x_{uv}\}_{\{u,v\} \in \binom{[n]}{2}}] / \langle \mathcal{P}_n \rangle_I,$$

with x_{uv} representing the indicator function of the edge uv being contained in a perfect matching.

Now we formulate low-degree derivations. Let \mathcal{P} denote a set of polynomials in $\mathbb{R}[x]$. For polynomials F and G , we write $F \simeq_{(\mathcal{P}, d)} G$, or F is congruent to G from \mathcal{P} in degree d , if and only if there exist polynomials $\{q(p) : p \in \mathcal{P}\}$ such that

$$F + \sum_{p \in \mathcal{P}} q(p) \cdot p = G$$

and $\max_p \deg(q(p) \cdot p) \leq d$. We often drop the dependence on \mathcal{P} when it is clear from context. We shall write $F \equiv G$ for two polynomials F and G defining the same function on perfect matchings, i.e., $F - G \in \langle \mathcal{P}_n \rangle_I$.

4.1.3 Deriving that symmetrized polynomials are constant

Averaging any polynomial on matchings over the symmetric group gives a constant. In this section we show that this fact has a low degree derivation.

For a partial matching M , let $x_M := \prod_{e \in M} x_e$ denote the product of edge variables for the edges in M . The first step is to reduce every polynomial to a linear combination of the x_M .

Lemma 61. *For every polynomial F there is a polynomial F' with $\deg F' \leq \deg F$ and $F \simeq_{(\mathcal{P}_n, \deg F)} F'$, where all monomials of F' have the form x_M for some partial matching M .*

Proof. It suffices to prove the lemma when F is a monomial. Let $F = \prod_{e \in A} x_e^{k_e}$ for

a set A of edges with multiplicities $k_e \geq 1$. From $x_e^2 \simeq_2 x_e$ it follows that $x_e^k \simeq_k x_e$ for all $k \geq 1$, hence $F \simeq_{\deg F} \prod_{e \in A} x_e$. If A is a partial matching we are done, otherwise there are distinct $e, f \in A$ with a common vertex, hence $x_e x_f \simeq_2 0$ and $F \simeq_{\deg F} 0$. \square

Lemma 62. *For any partial matching M on $2d$ vertices and a vertex b not covered by M , we have*

$$x_M \simeq_{(\mathcal{P}_n, d+1)} \sum_{\substack{M_1 = M \cup \{b, u\} \\ u \in K_n \setminus (M \cup \{b\})}} x_{M_1}. \quad (4.2)$$

Proof. We use the generators $\sum_u x_{bu} - 1$ to add variables corresponding to edges at b , and then use $x_{bu} x_{uv}$ to remove monomials not corresponding to a partial matching:

$$x_M \simeq_{(\mathcal{P}_n, d+1)} x_M \sum_{u \in K_n} x_{bu} \simeq_{(\mathcal{P}_n, d+1)} \sum_{\substack{M_1 = M \cup \{b, u\} \\ u \in K_n \setminus (M \cup \{b\})}} x_{M_1}.$$

\square

This leads to a similar congruence using all containing matchings of a larger size:

Lemma 63. *For any partial matching M of $2d$ vertices and $d \leq k \leq n/2$, we have*

$$x_M \simeq_{(\mathcal{P}_n, k)} \frac{1}{\binom{n/2-d}{k-d}} \sum_{\substack{M' \supset M \\ |M'|=k}} x_{M'}. \quad (4.3)$$

Proof. We use induction on $k - d$. The start of the induction is with $k = d$, when the sides of (4.3) are actually equal. If $k > d$, let b be a fixed vertex not covered by M . Applying Lemma 62 to M and b followed by the inductive hypothesis gives

$$x_M \simeq_{(\mathcal{P}_n, d+1)} \sum_{\substack{M_1 = M \cup \{b, u\} \\ u \in K_n \setminus (M \cup \{b\})}} x_{M_1} \simeq_{(\mathcal{P}_n, k)} \frac{1}{\binom{n/2-d-1}{k-d-1}} \sum_{\substack{M' \supset M_1 \\ |M'|=k \\ M_1 = M \cup \{b, u\} \\ u \in K_n \setminus (M \cup \{b\})}} x_{M'}.$$

Averaging over all vertices b not covered by M , we derive

$$\begin{aligned}
x_M &\simeq_{(\mathcal{P}_n, k)} \frac{1}{n-2d} \frac{1}{\binom{n/2-d-1}{k-d-1}} \sum_{\substack{M' \supset M_1 \\ |M'|=k \\ M_1 = M \cup \{b, u\} \\ b, u \in K_n \setminus M}} x_{M'} \\
&= \frac{1}{n-2d} \frac{1}{\binom{n/2-d-1}{k-d-1}} 2(k-d) \sum_{\substack{M' \supset M \\ |M'|=k}} x_{M'} \\
&= \frac{1}{\binom{n/2-d}{k-d}} \sum_{\substack{M' \supset M \\ |M'|=k}} x_{M'},
\end{aligned}$$

where in the second step the factor $2(k-d)$ accounts for the number of ways to choose b and u . □

We are now ready to state and prove the claim about symmetrized polynomials.

Lemma 64. *For any polynomial F , there is a constant c_F such that*

$$\sum_{\sigma \in S_n} \sigma F \simeq_{(\mathcal{P}_n, \deg F)} c_F.$$

Proof. Given Lemma 61, it suffices to prove the claim for $F = x_M$ for some partial matching M . Note that if $|M| = k$ the size of the stabilizer of M is $2^k k! (n-2k)!$. Now apply Lemma 63 with $d = 0$:

$$\sum_{\sigma \in S_n} \sigma x_M = 2^k k! (n-2k)! \sum_{M': |M'|=k} x_{M'} \simeq_k 2^k k! (n-2k)! \binom{n/2}{k}.$$

□

4.1.4 Low-degree certificates for matching ideal membership

In this section we present a crucial part of our argument, namely that every degree d polynomial that is identically zero over perfect matchings has a degree $O(d)$ derivation

of this fact.

The following lemma will allow us to apply induction:

Lemma 65. *If L is a polynomial with $L \simeq_{(\mathcal{P}_{n-2}, d)} 0$ for some d , and a, b are the two additional vertices in K_n , then $Lx_{ab} \simeq_{(\mathcal{P}_n, d+1)} 0$.*

Proof. It is enough to prove the claim for $L \in \mathcal{P}_{n-2}$. For $L = x_e^2 - x_e$ and $L = x_{uv}x_{uv}$ the claim is trivial since $L \in \mathcal{P}_n$ also. The remaining case is $L = \sum_{u \in K_{n-2}} x_{uv} - 1$ for some $v \in K_{n-2}$, in which case

$$Lx_{ab} = \left(\sum_{u \in K_n} x_{uv} - 1 \right) x_{ab} - x_{av}x_{ab} - x_{bv}x_{ab} \simeq_{d+1} 0.$$

□

We now show that any $F \in \langle \mathcal{P}_n \rangle_I$ can be generated by low-degree coefficients from \mathcal{P}_n .

Theorem 66. *For every polynomial $F \in \mathbb{R}[\{x_{uv}\}_{\{u,v\} \in \binom{[n]}{2}}]$, if $F \in \langle \mathcal{P}_n \rangle_I$ then*

$$F \simeq_{(\mathcal{P}_n, 2 \deg F - 1)} 0.$$

Proof. We use induction on the degree d of F . If $d = 0$ then $F = 0$ and the statement holds trivially. (Note that \simeq_{-1} is just equality.) The case $d = 1$ rephrased means that the affine space spanned by the characteristic vectors of all perfect matchings is defined by the $\sum_v x_{uv} - 1$ for all vertices u . This follows from Edmonds's description of the perfect matching polytope by linear inequalities in Edmonds [1965].

For the case $d \geq 2$ we first prove the following claim:

Claim. If $F \in \langle \mathcal{P}_n \rangle_I$ is a degree d polynomial and $\sigma \in S_n$ is a permutation of vertices, then

$$F \simeq_{(\mathcal{P}_n, 2d-1)} \sigma F.$$

First note that since $F \in \langle \mathcal{P}_n \rangle_I$, F is 0 on perfect matchings. Since σ simply permutes matchings, σF is also 0 on perfect matchings. It follows that $F - \sigma F \equiv 0 \pmod{\langle \mathcal{P}_n \rangle_I}$. The claim simply states that this identity is derivable within degree $2d - 1$.

To prove the claim we use induction on the degree. If $d = 0$ or $d = 1$ the claim follows from the corresponding cases $d = 0$ and $d = 1$ of the theorem. For $d \geq 2$ it is enough to prove the claim when σ is a transposition of two vertices a and u , since every permutation is a product of transpositions and chaining derivations does not increase the degree. Note that in $F - \sigma F$ all monomials which are independent of both a and u cancel:

$$F - \sigma F = \sum_{e: a \in e \text{ or } u \in e} L_e x_e, \quad (4.4)$$

where each L_e has degree at most $d - 1$. We now show that every summand is congruent to a sum of monomials containing edges incident to both a and u . For example, for $e = \{a, b\}$ in (4.4) we apply the generator $\sum_v x_{uv} - 1$ to find that

$$L_{ab} x_{ab} \simeq_{d+1} L_{ab} x_{ab} \sum_v x_{uv} \simeq_{d+1} \sum_v L_{ab} x_{ab} x_{uv}.$$

Therefore

$$F - \sigma F \simeq_{d+1} \sum_{bv} L'_{bv} x_{ab} x_{uv} \quad (4.5)$$

for some polynomials L'_{bv} of degree at most $d - 1$. We may assume that L'_{bv} does not contain variables x_e with e incident to a, b, u, v , as these can be removed using generators like $x_{ab} x_{ac}$ or $x_{ab}^2 - x_{ab}$.

As stated earlier, the left hand side of (4.5) is 0 on perfect matchings, so the right hand side is also. We now show that not just the sum but in fact each summand on the right hand side of (4.5) is 0 on perfect matchings. Fix b and v and consider the summand $L'_{bv} x_{ab} x_{uv}$. This term is 0 on any perfect matching not containing both ab and uv , so we just need to show that L'_{bv} is 0 on all perfect matchings containing both

ab and uv . Note that any other summand $L'_{b'v'}x_{ab'}x_{uv'}$ is 0 on any perfect matching containing ab and uv since either $b \neq b'$ or $v \neq v'$. Since on all perfect matchings containing ab and uv both the left hand side is 0 and every other summand on the right hand side is 0, it follows that L'_{bv} is also.

To complete the proof we need to show that this fact is derivable in degree $2d - 1$ (note that $2d - 1 \geq d + 1$ for $d \geq 2$). Formally, for each b and v we show that $L'_{bv}x_{ab}x_{uv} \simeq_{2d-1} 0$. We only need to consider the choices of b and v such that ab and uv are part of a perfect matching:

1. If $b = u$ and $v = a$ we have $L'_{au}x_{au}x_{au} \simeq_{d+1} L'_{au}x_{au}$, and as shown before, L'_{au} is 0 on all perfect matchings containing au . Thus $L'_{au} \in \langle \mathcal{P}_{n-2} \rangle_I$, if we identify a, u as the two additional vertices in K_n . By induction we have $L'_{au} \simeq_{2d-3} 0$ and applying Lemma 65 we conclude $L'_{au}x_{au} \simeq_{2d-2} 0$.
2. If a, b, u, v are distinct we have that L'_{bv} is 0 on all perfect matchings containing ab and uv . Thus $L'_{bv} \in \langle \mathcal{P}_{n-4} \rangle_I$, if we identify a, b, u, v as the four additional vertices in K_n . By induction we have $L'_{bv} \simeq_{2d-3} 0$ and by applying Lemma 65 twice we conclude $L'_{bv}x_{ab}x_{uv} \simeq_{2d-1} 0$.

This concludes the proof of the claim.

We now apply the claim followed by Lemma 64 to derive that

$$F \simeq_{2d-1} \frac{1}{n!} \sum_{\sigma \in S_n} \sigma F \simeq_d \frac{c_F}{n!}$$

for some constant c_F . As $F \in \langle \mathcal{P}_n \rangle_I$, it must be that $c_F = 0$ and therefore

$$F \simeq_{2d-1} 0.$$

□

4.1.5 The symmetric SDP lower bound

We now have all the ingredients to prove our lower bound. Note that the alternating group A_n acts naturally on $\text{PM}(n)$ via permutation of vertices, and the guarantees \tilde{C}, \tilde{S} are A_n -symmetric. Our theorem is an exponential lower bound on the size of any A_n -coordinate-symmetric SDP extension of $\text{PM}(n)$.

Theorem 67. *There exists a constant $\alpha > 0$ such that for all even n and every $0 \leq \varepsilon < 1$, every A_n -coordinate-symmetric (\tilde{C}, \tilde{S}) -approximate SDP extended formulation for the perfect matching problem $\text{PM}(n)$ has size at least $2^{\alpha n}$. In particular, every A_n -coordinate-symmetric SDP extended formulation approximating the perfect matching problem $\text{PM}(n)$ within a factor of $1 - \varepsilon/(n - 1)$ has size at least $2^{\alpha n}$.*

Proof. Fix an even integer $n \geq 10$ and let $k = \lceil \beta n \rceil$ for some small enough constant $0 < \beta < 1/2$ chosen later. Suppose for a contradiction that $\text{PM}(n)$ admits a symmetric SDP extended formulation of size $d < \sqrt{\binom{n}{k}} - 1$.

Let m equal $n/2$ or $n/2 - 1$, whichever is odd. Let $S = [m]$ and let $T = \{m + 1, \dots, 2m\}$. If $m = n/2$ then let $U = \{2m + 1, 2m + 2\}$, otherwise let $U = \emptyset$. Note that $S \cup T \cup U = [n]$ and $|S| = |T| = m = \Theta(n)$. Consider the objective function for the set of edges $E[S]$ on S . Since $|S|$ is odd we have $\max f_{E[S]} = (|S| - 1)/2$, from which we obtain

$$f(x) \stackrel{\text{def}}{=} \tilde{C}(f_{E[S]}) - f_{E[S]}(x) \tag{4.6}$$

$$= \frac{|S| - 1}{2} + \frac{\varepsilon}{2} - \sum_{u,v \in S} x_{uv} \tag{4.7}$$

$$\equiv \frac{1}{2} \sum_{u \in S, v \in T \cup U} x_{uv} - \frac{1 - \varepsilon}{2}. \tag{4.8}$$

By Lemma 57, as $\binom{d+1}{2} < \binom{n}{k}$, there is a constant $\mu_f \geq 0$ and an A_n -symmetric set

\mathcal{H} of functions of size at most $\binom{n}{k}$ on the set of perfect matchings such that

$$f \equiv \sum_g g^2 + \mu_f$$

where each $g \in \langle \mathcal{H} \rangle$. By Proposition 60, every $h \in \mathcal{H}$ depends only on the edges within a vertex set of size less than k , and hence can be represented by a polynomial of degree less than $k/2$ over perfect matchings. As the g are linear combinations of the $h \in \mathcal{H}$, they can also be represented by polynomials of degree less than $k/2$, which we assume for the rest of the proof.

Applying Theorem 66 with (4.6), we conclude

$$\frac{1}{2} \sum_{u \in S, v \in T \cup U} x_{uv} - \frac{1 - \varepsilon}{2} \simeq_{(\mathcal{P}_n, 2k-1)} \sum_g g^2 + \mu_f.$$

We now apply the following substitution: set $x_{2m+1, 2m+2} := 1$ if U is not empty, set $x_{u+m, v+m} := x_{uv}$ for each $uv \in E[S]$, and set $x_{uv} := 0$ otherwise. Intuitively, the substitution ensures that U is matched, ensures the matching on T is identical to the matching on S , and ensures every edge is entirely within S , T , or U . The main point is that the substitution maps every polynomial in \mathcal{P}_n either to 0 or into \mathcal{P}_m .

Applying this substitution we obtain a new polynomial identity on the variables

$$\{x_{uv}\}_{\{u,v\} \in \binom{S}{2}}: \quad -\frac{1 - \varepsilon}{2} \simeq_{(\mathcal{P}_m, 2k-1)} \sum_g g^2 + \mu_f. \quad (4.9)$$

(4.9) is a sum of squares refutation of the existence of a perfect matching in an odd clique of size m . We are now ready to apply the following theorem.

Theorem 68 ([Grigoriev, 2001, Corollary 2]). *The degree of any $PC_{>}$ refutation of MOD_2^k is greater than $\Omega(k)$.*

The MOD_p^k principle states that it is not possible to partition a set of size k into groups of size p if k is congruent to 1 modulo p . In our case, with $p = 2$ and k odd,

this is equivalent to the statement that no perfect matching exists in an odd clique. It can also be checked that (4.9) constitutes a $PC_{>}$ refutation; see [Grigoriev, 2001, Definition 2], also see Buss et al. [1999] for further discussion.

By applying Theorem 68 to (4.9), it follows that $2k - 1 = \Omega(m) = \Omega(n)$, a contradiction when β is chosen small enough. \square

4.2 A note on the inapproximability ratio

The inapproximability ratio claimed in our theorem is $1 - \frac{\epsilon}{n-1}$. To be precise, the actual ratio implied by our argument is $1/(1 + \frac{2\epsilon}{n-4})$. In other words, we actually show that any small symmetric SDP cannot achieve even a (slightly) worse ratio.

Here we derive the actual ratio implied by the argument. In the setup, $n \geq 10$ is an even integer and $m = n/2$ or $n/2 - 1$ (whichever is odd). Let us consider the case $m = n/2 - 1$ since this gives the worse ratio.

We consider maximum matchings over $S = [m]$. Let

$$f(M) := f_{E[S]}(M) = |M \cap S|$$

for any perfect matching M . We have $\tilde{S}(f) := \max f = \frac{m-1}{2} = \frac{n-4}{4}$ and $\tilde{C}(f) := \max f + \frac{\epsilon}{2}$ for some $0 \leq \epsilon < 1$. We show that a small symmetric SDP cannot derive $\max f \leq \tilde{C}(f)$ when $\max f \leq \tilde{S}(f)$ and thus cannot achieve a ρ approximation, where

$$\frac{1}{\rho} = \frac{\tilde{C}(f)}{\tilde{S}(f)} = \frac{\max f + \frac{\epsilon}{2}}{\max f} = 1 + \frac{\frac{\epsilon}{2}}{\frac{n-4}{4}} = 1 + \frac{2\epsilon}{n-4}.$$

Note that for $n > 4$ and $0 \leq \epsilon < 1$ we have

$$\frac{1}{1 + \frac{2\epsilon}{n-4}} \leq 1 - \frac{\epsilon}{n-1}.$$

CHAPTER 5

SMALL SYMMETRIC CP FORMULATIONS

We now consider optimization over the copositive cone and its dual, the completely positive cone. We will refer to any formulation in this framework as a copositive (CP) formulation, even though the geometric object corresponding to the feasible region may itself lie in the completely positive cone.

We will define the concept of a symmetric CNF and extend the work of Maksimenko [2012] and Fiorini et al. [2015a] to show that any problem whose feasible solutions can be expressed by a symmetric CNF has a small copositive formulation. Finally, we give a symmetric CNF for the matching problem, thus establishing a small copositive formulation for matching.

5.1 A symmetric CP extension for the correlation polytope

Fiorini et al. [2015a] gave a small copositive extension for the correlation polytope. We now analyze this extension to show it is in fact symmetric.

Consider the matrices $Y \in \mathcal{C}_{1+2n}^*$ that satisfy the following conditions, where the matrix indices range from 0 to $2n$:

$$\begin{aligned} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} Y = 1, \\ \begin{bmatrix} 0 & e_i^\top & e_i^\top \\ e_i & 0 & 0 \\ e_i & 0 & 0 \end{bmatrix} Y = 2 \quad i = 1, \dots, n, \end{aligned}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & e_i e_i^\top & e_i e_i^\top \\ 0 & e_i e_i^\top & e_i e_i^\top \end{bmatrix} Y = 1 \quad i = 1, \dots, n,$$

and

$$\begin{bmatrix} 0 & e_j^\top & 0 \\ e_j & -2e_j e_j^\top & 0 \\ 0 & 0 & 0 \end{bmatrix} Y = 0 \quad j = 1, \dots, n.$$

Note that the coefficients of Y are block matrices whose diagonal elements have dimensions 1×1 , $n \times n$, and $n \times n$ respectively. Let \mathcal{Y} denote the set of matrices $Y \in \mathcal{C}_{1+2n}^*$ that satisfy all the conditions above. Observe that \mathcal{Y} is an affine slice of the completely positive cone \mathcal{C}_{1+2n}^* .

The definition of the set \mathcal{Y} is taken from [Fiorini et al., 2015a], who proved the following theorem.

Theorem 69 ([Fiorini et al., 2015a]). *The set \mathcal{Y} is a polynomial size completely positive extension of the correlation polytope. In particular,*

$$\text{COR}(n) = \{Z \in \mathbb{R}^{n \times n} \mid \exists Y \in \mathcal{Y} : Z_{ij} = Y_{ij}, \forall i, j = 1, \dots, n\}.$$

To derive the results of this section we make the following crucial observation.

Theorem 70. *The set \mathcal{Y} is an S_n -symmetric extension of the correlation polytope.*

Proof. It suffices to give the action of S_n on $\mathbb{R}^{(1+2n) \times (1+2n)}$. Let $\sigma : [n] \rightarrow [n]$ be an element of S_n . Define the action of σ on the set $\{0, \dots, 2n\}$ by

$$\sigma(i) = \begin{cases} 0 & i = 0, \\ \sigma(i) & i \in \{1, \dots, n\}, \\ n + \sigma(i - n) & i \in \{n + 1, \dots, 2n\}, \end{cases}$$

and let σ simultaneously permute the rows and columns of an element of $\mathbb{R}^{(1+2n) \times (1+2n)}$ according to its action on $\{0, \dots, 2n\}$. \square

5.2 Connecting the cut and correlation polytopes

In order to connect the cut and correlation polytopes we will make use of Theorem 40, which we restate here.

Theorem ([De Simone, 1989/90]). *For all n , $\text{COR}(n)$ is linearly isomorphic to $\text{CUT}(1+n)$.*

If we consider the graph K_{1+n} , on which $\text{CUT}(1+n)$ is based, and label its vertices from 0 to n (where 0 is a special designated vertex), then the mapping from $X \in \mathbb{R}^{n \times n}$ to $\delta \in \mathbb{R}^{\binom{1+n}{2}}$ is given by

$$\delta_{0i} := X_{ii}$$

for $i \in [n]$ and

$$\delta_{ij} := X_{ii} - X_{ij} + X_{jj} - X_{ji}$$

for $1 \leq i < j \leq n$.

The inverse mapping is given by

$$X_{ii} := \delta_{0i}$$

for $1 \leq i \leq n$ and

$$X_{ij} := \frac{1}{2}(\delta_{0i} + \delta_{0j} - \delta_{ij})$$

for $i, j \in [n]$ and $i \neq j$.

5.3 Maksimenko's construction

Our formulation relies critically on the construction given by Maksimenko [2012] which we describe here. We first introduce some terminology. The *length* of a CNF ϕ , denoted $|\phi|$, is the sum of the lengths of its clauses, where the length of a clause is the number of literals it contains. If ϕ is a CNF in variables x_1, \dots, x_k , the set $\text{SAT}(\phi)$ consists of the strings $x = (x_1, \dots, x_k) \in \{0, 1\}^k$ that satisfy ϕ . We can now define the following object:

Definition 71 (polytope of a formula). Let ϕ be a CNF in k variables. The *polytope of ϕ* is defined as

$$P(\phi) := \text{conv} \{x \in \{0, 1\}^k \mid x \in \text{SAT}(\phi)\}.$$

Note that $P(\phi)$ is a 0/1 polytope in \mathbb{R}^k .

We now state Maksimenko's theorem.

Theorem 72 (Maksimenko [2012]). *There is a polynomial p such that for any CNF ϕ the polytope $P(\phi)$ is an orthogonal projection of a face of $\text{CUT}(d)$ for some $d \leq p(|\phi|)$.*

Here we recap the construction given by Maksimenko. Let ϕ have k variables and m clauses C_1, \dots, C_m . To avoid degenerate cases we assume that ϕ is satisfiable and that every clause contains at least two literals.

For every $i \in [m]$ we define a vertex set V_i consisting of a vertex $v(a, C_i)$ for each literal a that appears in C_i . In addition for each $j \in [k]$ we have a vertex v_j and a vertex \bar{v}_j corresponding to each variable and its negation. Let

$$V := \left(\bigcup_{i \in [m]} V_i \right) \cup \left(\bigcup_{j \in [k]} \{v_j, \bar{v}_j\} \right)$$

and let $n = |V|$. We create an additional root vertex v_0 and consider the cut problem on the complete graph with vertex set $V' := \{v_0\} \cup V$. In other words, we consider the polytope $\text{CUT}(1+n)$.

For any $v, w \in V'$ let $x(v, w)$ denote the corresponding edge variable in $\mathbb{R}^{\binom{1+n}{2}}$. Let $L(\phi)$ be the affine subspace of $\mathbb{R}^{\binom{1+n}{2}}$ defined by the following equations.

1. For each $j \in [k]$:

$$x(v_j, \bar{v}_j) - x(v_0, v_j) - x(v_0, \bar{v}_j) = 0.$$

2. For each $i \in [m]$:

$$\sum_{\substack{v, w \in V_i \\ v \neq w}} x(v, w) - (|V_i| - 2) \sum_{v \in V_i} x(v_0, v) = 1.$$

The relevant face of $\text{CUT}(1+n)$ is obtained by intersecting with $L(\phi)$. Lastly, the projection onto $P(\phi)$ in \mathbb{R}^k is given by $x_j = x(v_0, v_j)$ for $j \in [k]$.

Corollary 73. *For any CNF ϕ , the polytope $P(\phi)$ has a polynomial size (in $|\phi|$) completely positive extension.*

Proof. By Theorem 72 there is a d polynomial in the length of ϕ and an affine space $L = L(\phi) \subseteq \mathbb{R}^{\binom{d}{2}}$ such that $P(\phi)$ is the projection of $L \cap \text{CUT}(d)$. Let Λ denote the affine isomorphism from $\text{COR}(d-1)$ to $\text{CUT}(d)$. It follows that $P(\phi)$ is an affine projection of $\Lambda^{-1}L \cap \text{COR}(d-1)$. The claim follows by Theorem 69. \square

Corollary 74. *Every language in NP/poly has a polynomial size completely positive extension.*

5.4 Symmetric CNFs

Let G be a group acting on $[k]$. If ϕ is a CNF on k variables, the action of G on $[k]$ induces an action on ϕ by permuting the variables. Specifically, the action of $g \in G$

on ϕ is to replace each occurrence of the variable x_j in ϕ with $x_{g \cdot j}$, for every $j \in [k]$.

Definition 75 (symmetric CNF). Let ϕ be a CNF on k variables and let G be a group acting on $[k]$. The formula ϕ is *G-symmetric* if the action of G on ϕ leaves ϕ unchanged, up to reordering of clauses and reordering of literals within each clause.

Observation 76. If ϕ is a CNF, G is a group, and ϕ is G -symmetric, then the polytope $P(\phi)$ is also G -symmetric.

Proposition 77. *If ϕ is a G -symmetric CNF, the affine subspace $L(\phi)$ is also G -symmetric.*

In both Observation 76 and Proposition 77, the action of G on \mathbb{R}^k is the induced action on coordinates given by the action of G on the variables x_j for $j \in [k]$.

We are now ready to state our main lemma.

Lemma 78. *If G is a group acting on $[k]$ and ϕ is a G -symmetric CNF, then the polytope $P(\phi)$ has a G -symmetric completely positive extension with size polynomial in $|\phi|$.*

Proof. We simply need to show that the polynomial size completely positive extension given by Corollary 73 can be constructed to preserve G -symmetry. In particular, it suffices to define and check the action of G on each component of the construction.

1. By Proposition 77, the affine space $L = L(\phi)$ is G -symmetric, where the action of G on L follows naturally from the action of G on ϕ .
2. The action of G on ϕ naturally defines an action on V' , the vertex set of the graph corresponding to $\text{CUT}(d)$ that leaves V' invariant and fixes v_0 .
3. The action of G on V' induces an action on the coordinates of $\mathbb{R}^{\binom{d}{2}}$ that leaves $\text{CUT}(d)$ invariant and fixes the role of v_0 .

4. The action of G on $\text{CUT}(d)$, combined with the isomorphism Λ from $\text{COR}(d-1)$ to $\text{CUT}(d)$ implies an action of G on the coordinates of $\mathbb{R}^{(d-1) \times (d-1)}$ that involves simultaneous permutation of rows and columns and thus leaves $\text{COR}(d-1)$ invariant.
5. The action of G on $L(\phi)$ induces an action on $\Lambda^{-1}L$ that leaves it invariant.

It follows that the set $\Lambda^{-1}L \cap \text{COR}(d-1)$ is G -symmetric. It can be checked that the polynomial size completely positive extension of this set implied by Theorem 69 is also G -symmetric. It is easy to verify that $g \cdot \pi(x) = \pi(g \cdot x)$ holds for any $g \in G$ and x in this extension. \square

5.5 A small symmetric CP formulation for matching

Consider the complete graph K_n on n vertices, with n even. A perfect matching on K_n is a vertex-disjoint edge cover of $[n]$. We can view the perfect matchings on K_n as elements of $\mathbb{R}^{\binom{n}{2}}$, where each perfect matching is represented by its characteristic vector. Let the perfect matching polytope be defined as

$$\text{PM}(n) := \text{conv} \left\{ x \in \mathbb{R}^{\binom{n}{2}} \mid x \text{ is a perfect matching} \right\}.$$

The following proposition establishes that the matching polytope is symmetric.

Proposition 79. *PM(n) is S_n -symmetric, where the action on $\mathbb{R}^{\binom{n}{2}}$ is induced naturally by the action on the vertex set $[n]$.*

For each n , define

$$\phi_n := \left(\bigwedge_{i \in [n]} \bigvee_{\substack{j \in [n] \\ j \neq i}} x_{ij} \right) \wedge \left(\bigwedge_{i \in [n]} \bigwedge_{\substack{j, k \in [n] \\ i, j, k \text{ distinct}}} (\overline{x_{ij}} \vee \overline{x_{ik}}) \right).$$

The following proposition establishes the relationship between the matching polytope and the CNF just described.

Proposition 80.

$$\text{PM}(n) = P(\phi_n).$$

We are now ready to state and prove our main theorem.

Theorem 81. *PM(n) has a polynomial size S_n -symmetric completely positive extension.*

Proof. It is easy to check that $|\phi_n|$ is polynomial in n and that ϕ_n is S_n -symmetric.

The claim follows by Proposition 80 and Lemma 78. □

CHAPTER 6

A SMALL SYMMETRIC CP FORMULATION FOR TSP

In light of the framework presented in Chapter 5, in order to give a small symmetric completely positive extended formulation for the traveling salesperson problem, it suffices to exhibit a small symmetric CNF. To do this, we will view a tour of the complete graph simultaneously as:

1. a permutation of its vertices, and
2. a subset of its edges.

We will construct a small CNF formula where each satisfying assignment encodes both representations of a particular tour. As part of the construction we will use the fact that a tour of the complete graph is also a tour of the complete directed graph of the same size.

We will then show that our formula is invariant under an appropriately defined action of the symmetric group. Projecting onto the variables corresponding to edges will recover the characteristic vectors of Hamiltonian cycles in the complete graph.

6.1 The construction

Fix $n \in \mathbb{N}$. As before, K_n denotes the complete undirected graph whose vertex set is $[n]$. Let \vec{K}_n denote the complete directed graph whose vertex set is $[n]$.

In the following, the indices i, j, u, v , and w all take values in $[n]$, however i and j represent positions in the tour (e.g. the i th city visited) whereas u, v , and w indicate vertices of K_n or \vec{K}_n .

6.1.1 Variables

We will represent a tour of K_n as a permutation $\sigma \in S_n$ of its vertices, where $\sigma(i)$ is the i th vertex in the tour. We will represent σ using the set of Boolean variables

$$\Sigma := \{\sigma_{iv} \mid i, v \in [n]\}$$

where σ_{iv} is true iff vertex v is the i th vertex visited.

Using the fact that σ also defines a tour of \vec{K}_n , we will represent the set of directed edges in \vec{K}_n corresponding to σ using the set of Boolean variables

$$Z := \{z_{uv} \mid u, v \in [n], u \neq v\}$$

where z_{uv} is true iff vertex v is visited immediately after vertex u .

Finally, we will represent the set of undirected edges in K_n corresponding to σ using the set of Boolean variables

$$X := \{x_{uv} \mid u, v \in [n], u < v\}$$

where x_{uv} is true iff u and v are adjacent in the tour corresponding to σ .

6.1.2 Encoding a permutation

The constraints listed below ensure that Σ encodes a permutation of $[n]$.

$$\bigvee_{v \in [n]} \sigma_{iv} \quad i \in [n] \quad (6.1)$$

$$\bigwedge_{\substack{u, v \in [n] \\ u \neq v}} \overline{\sigma_{iu}} \vee \overline{\sigma_{iv}} \quad i \in [n] \quad (6.2)$$

$$\bigvee_{i \in [n]} \sigma_{iv} \quad v \in [n] \quad (6.3)$$

$$\bigwedge_{\substack{i, j \in [n] \\ i \neq j}} \overline{\sigma_{iv}} \vee \overline{\sigma_{jv}} \quad v \in [n] \quad (6.4)$$

Informally, (6.1) ensures that the slot for $\sigma(i)$ is assigned to a vertex, while (6.2) ensures it is not multiply assigned. Similarly, (6.3) ensures vertex v is visited, while (6.4) ensures it is not visited more than once.

6.1.3 Encoding cycles

The constraints below ensure that Z is a disjoint cycle cover of \vec{K}_n .

$$\bigvee_{v \in [n]} z_{uv} \quad u \in [n] \quad (6.5)$$

$$\bigwedge_{\substack{v, w \in [n] \\ v \neq w}} \overline{z_{uv}} \vee \overline{z_{uw}} \quad u \in [n] \quad (6.6)$$

$$\bigvee_{u \in [n]} z_{uv} \quad v \in [n] \quad (6.7)$$

$$\bigwedge_{\substack{u, w \in [n] \\ u \neq w}} \overline{z_{uv}} \vee \overline{z_{vw}} \quad v \in [n] \quad (6.8)$$

Informally, (6.5) ensures that vertex u has an outgoing edge, while (6.6) ensures it does not have multiple outgoing edges. Similarly, (6.7) ensures vertex v has an

incoming edge, while (6.8) ensures it does not have multiple incoming edges.

6.1.4 From a permutation to a directed tour

For an index variable i let i' denote $i + 1 \pmod{n}$, with the appropriate adjustment for 1-based indexing. The following set of constraints ensures that Z conforms to Σ :

$$\bigwedge_{i \in [n]} (\overline{\sigma_{iu}} \vee \overline{\sigma_{i'v}} \vee z_{uv}) \quad u, v \in [n], u \neq v. \quad (6.9)$$

Informally, (6.9) encodes

$$\left[\bigvee_{i \in [n]} (\sigma_{iu} \wedge \sigma_{i'v}) \right] \rightarrow z_{uv}, \quad (*)$$

which says that directed edge uv is in Z if v immediately follows u in σ .

6.1.5 From a directed tour to an undirected tour

The constraints below ensure that X is the undirected version of Z .

$$\overline{x_{uv}} \vee z_{uv} \vee z_{vu} \quad u, v \in [n], u < v \quad (6.10)$$

$$\overline{z_{uv}} \vee x_{uv} \quad u, v \in [n], u < v \quad (6.11)$$

$$\overline{z_{vu}} \vee x_{uv} \quad u, v \in [n], u < v \quad (6.12)$$

Informally, (6.10) encodes

$$x_{uv} \rightarrow (z_{uv} \vee z_{vu})$$

(every undirected edge has a directed counterpart), while (6.11) and (6.12) encode

$$z_{uv} \rightarrow x_{uv}$$

$$z_{vu} \rightarrow x_{uv}$$

(every directed edge has an undirected counterpart).

Let $\Phi(\Sigma, Z, X)$ denote the CNF that consists of the AND of constraints (6.1) through (6.12).

6.1.6 Defining a group action

The action of $\rho \in S_n$ on the variables (Σ, Z, X) is defined by the following maps.

$$\sigma_{iv} \rightarrow \sigma_{i\rho(v)} \tag{6.13}$$

$$z_{uv} \rightarrow z_{\rho(u)\rho(v)} \tag{6.14}$$

$$x_{uv} \rightarrow x_{\rho(u)\rho(v)}. \tag{6.15}$$

6.1.7 Putting it all together

We can now state the main theorem of this chapter:

Theorem 82. *The TSP problem has a small symmetric copositive extension.*

Proof. It is easy to check that every satisfying assignment of Φ corresponds to a TSP tour and every TSP tour is represented by a satisfying assignment. We can also verify that the action defined in (6.13)–(6.15) is consistent with the action on TSP tours induced by permuting vertices, and the projection of the satisfying assignments (Σ, Z, X) of Φ to X expresses exactly the characteristic vectors of TSP tours. Finally we note that Φ has size polynomial in n and is invariant under the group action. \square

CHAPTER 7

CONCLUSION

We have considered the role of symmetry in extended formulations. Generalizing the work of Yannakakis, we showed that the matching problem has no small symmetric semidefinite program. We then gave a framework for producing small symmetric copositive programs and showed that both matching and TSP have small copositive programs in this framework.

Several open questions remain. Most prominent: does the matching problem have a small semidefinite program if we allow asymmetry? An answer either way would fill in the last entry in Table 1.1 on page 22, and complete a line of research that extends back nearly 30 years to Yannakakis. If the answer is yes, it would be a strong example of the power of asymmetry in semidefinite extended formulations. If the answer is no, it would point to the need to find more powerful but still efficient models of computation.

Regardless of whether matching has a small asymmetric semidefinite program, the power of asymmetry in general is not well understood. Fawzi et al. [2014, 2015] show that in some cases asymmetry can help for semidefinite formulations. In contrast, we have given evidence that symmetry is not a strong restriction for copositive programs. Even though copositive programming is NP-hard, we note that it can still be useful to have a small symmetric copositive program. For example, Dobre and Vera [2015] show how symmetry in copositive programs can be exploited in SDP approximations.

Our symmetric copositive formulation for the TSP could possibly be generalized to other problems. We conclude by phrasing this possibility as an open question.

Open Question. *Does every symmetric set that has a small CNF also have a small symmetric CNF?*

APPENDIX A

PERMISSION TO REPRINT

The contents of Chapter 3 and Chapter 4 first appeared in an abridged form in

Gábor Braun, Jonah Brown-Cohen, Arefin Huq, Sebastian Pokutta, Prasad Raghavendra, Aurko Roy, Benjamin Weitz, and Daniel Zink. The matching problem has no small symmetric SDP. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1067–1078. SIAM, 2016. ISBN 978-1-61197-433-1. doi: 10.1137/1.9781611974331.ch75. URL <http://dx.doi.org/10.1137/1.9781611974331.ch75>

and appear in this dissertation with the kind permission of the Society for Industrial and Applied Mathematics as shown on the following page.



Arefin Huq <arefin@gmail.com>

Permission to reuse SODA paper in PhD thesis

2 messages

Arefin Huq <arefin@gatech.edu>
To: thomas@siam.org

Tue, Feb 9, 2016 at 2:27 PM

Dear Kelly Thomas,

I am a co-author of the paper linked below, which appeared in the proceedings of SODA '16.

<http://dx.doi.org/10.1137/1.9781611974331.ch75>

I would like to incorporate this paper into my PhD thesis at Georgia Tech and am hereby requesting permission to do so. I will of course give a proper citation of the paper and the proceedings.

Please let me know how to proceed.

Thanks,
A

Kelly Thomas <Thomas@siam.org>
To: Arefin Huq <arefin@gatech.edu>

Tue, Feb 9, 2016 at 3:06 PM

Dear Mr. Huq:

SIAM is happy to give permission to use the paper mentioned below. Please acknowledge the full bibliographic information for the original proceedings paper.

Sincerely,

Kelly Thomas

Managing Editor, SIAM

From: Arefin Huq [mailto:arefin@gatech.edu]
Sent: Tuesday, February 09, 2016 2:29 PM
To: Kelly Thomas
Subject: Permission to reuse SODA paper in PhD thesis

[Quoted text hidden]

REFERENCES

- Egon Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Appl. Math.*, 89:1–44, 1998. 19
- Abbas Bazzi, Samuel Fiorini, Sebastian Pokutta, and Ola Svensson. No small linear program approximates vertex cover within a factor $2 - \epsilon$. In *Proc. FOCS*, pages 1123–1142, 2015. 21
- William Blake. *Songs of Innocence and of Experience*. Author, 1794.
- Gábor Braun and Sebastian Pokutta. An algebraic take on symmetric extended formulations. Manuscript, 2011. 44
- Gábor Braun and Sebastian Pokutta. The matching polytope does not admit fully-polynomial size relaxation schemes. In *Proc. SODA*, pages 837–846, 2015a. 43
- Gábor Braun and Sebastian Pokutta. The matching problem has no fully polynomial size linear programming relaxation schemes. *IEEE Transactions on Information Theory*, 61(10):5754–5764, 2015b. doi: 10.1109/TIT.2015.2465864. URL <http://dx.doi.org/10.1109/TIT.2015.2465864>. 43
- Gábor Braun, Samuel Fiorini, Sebastian Pokutta, and David Steurer. Approximation limits of linear programs (beyond hierarchies). In *Proc. FOCS*, pages 480–489, 2012. 21
- Gábor Braun, Samuel Fiorini, Sebastian Pokutta, and David Steurer. Approximation limits of linear programs (beyond hierarchies). *Math. Oper. Res.*, 40(3):756–772, 2015a. doi: 10.1287/moor.2014.0694. URL <http://dx.doi.org/10.1287/moor.2014.0694>. 21
- Gábor Braun, Sebastian Pokutta, and Aurko Roy. Strong reductions for extended formulations. *CoRR*, abs/1512.04932, 2015b. URL <http://arxiv.org/abs/1512.04932>. 21
- Gábor Braun, Sebastian Pokutta, and Daniel Zink. Inapproximability of combinatorial problems via small lps and sdps. In *Proc. STOC*, pages 107–116, 2015c. 21, 36
- Gábor Braun, Jonah Brown-Cohen, Arefin Huq, Sebastian Pokutta, Prasad Raghavendra, Aurko Roy, Benjamin Weitz, and Daniel Zink. The matching problem has no small symmetric SDP. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1067–1078. SIAM, 2016. ISBN 978-1-61197-433-1. doi: 10.1137/1.9781611974331.ch75. URL <http://dx.doi.org/10.1137/1.9781611974331.ch75>. 19, 20, 36, 43

- Mark Braverman and Ankur Moitra. An information complexity approach to extended formulations. In *Proc. STOC*, pages 161–170, 2013. 21
- Jop Briët, Daniel Dadush, and Sebastian Pokutta. On the existence of 0/1 polytopes with high semidefinite extension complexity. *Proceedings of ESA / arXiv:1305.3268*, pages 217–228, 2013. 22
- Jop Briët, Daniel Dadush, and Sebastian Pokutta. On the existence of 0/1 polytopes with high semidefinite extension complexity. *Math. Program.*, 153(1):179–199, 2015. doi: 10.1007/s10107-014-0785-x. URL <http://dx.doi.org/10.1007/s10107-014-0785-x>. 22
- Arne Brøndsted. *An Introduction to Convex Polytopes*. Graduate texts in mathematics. Springer, 1983. 29
- Sam Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 547–556. ACM, 1999. 54
- Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 350–359. IEEE, 2013. 21, 22
- Clay Mathematics Institute. Millennium problems, 2016. URL <http://www.claymath.org/millennium-problems>. 2
- Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM. doi: 10.1145/800157.805047. URL <http://doi.acm.org/10.1145/800157.805047>. 2
- Caterina De Simone. The cut polytope and the Boolean quadric polytope. *Discrete Math.*, 79:71–75, 1989/90. doi: 10.1016/0012-365X(90)90056-N. 32, 57
- John D. Dixon and Brian Mortimer. *Permutation groups*. Springer Verlag, 1996. 44
- Cristian Dobre and Juan Vera. Exploiting symmetry in copositive programs via semidefinite hierarchies. *Mathematical Programming*, 151(2):659–680, 2015. ISSN 1436-4646. doi: 10.1007/s10107-015-0879-0. URL <http://dx.doi.org/10.1007/s10107-015-0879-0>. 18, 68
- Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *J. Res. Nat. Bur. Standards Sect. B*, 69B:125–130, 1965. ISSN 0160-1741. 19, 49
- Hamza Fawzi, James Saunderson, and Pablo A. Parrilo. Equivariant semidefinite lifts of regular polygons. *CoRR*, abs/1409.4379, 2014. URL <http://arxiv.org/abs/1409.4379>. 68

- Hamza Fawzi, James Saunderson, and Pablo A. Parrilo. Equivariant semidefinite lifts and sum-of-squares hierarchies. *SIAM Journal on Optimization*, 25(4):2212–2243, 2015. doi: 10.1137/140966265. URL <http://dx.doi.org/10.1137/140966265>. 22, 68
- Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Linear vs. semidefinite extended formulations: Exponential separation and strong lower bounds. *Proceedings of STOC*, pages 95–106, 2012. 21, 22
- Samuel Fiorini, Serge Massar, Manas K Patra, and Hans Raj Tiwary. Generalized probabilistic theories and conic extensions of polytopes. *Journal of Physics A: Mathematical and Theoretical*, 48(2):025302, 2015a. URL <http://stacks.iop.org/1751-8121/48/i=2/a=025302>. 23, 55, 56
- Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *J. ACM*, 62(2):17, 2015b. doi: 10.1145/2716307. URL <http://doi.acm.org/10.1145/2716307>. 21, 22
- Michel X. Goemans. Smallest compact formulation for the permutahedron. *Math. Program.*, 153(1):5–11, 2015. doi: 10.1007/s10107-014-0757-1. URL <http://dx.doi.org/10.1007/s10107-014-0757-1>. 12, 21
- Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995. doi: 10.1145/227683.227684. 22
- João Gouveia, Parrilo A. Parrilo, and Rekha R. Thomas. Lifts of convex sets and cone factorizations. *Math. Oper. Res.*, 38(2):248–264, May 2011. 22
- Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1):613–622, 2001. 20, 53, 54
- Volker Kaibel, Kanstantsin Pashkovich, and Dirk Oliver Theis. Symmetry matters for the sizes of extended formulations. In *Proc. IPCO 2010*, pages 135–148, 2010. 21, 44
- Monique Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming. *Math. Oper. Res.*, pages 470–496, 2003. 17
- James R. Lee, Prasad Raghavendra, David Steurer, and Ning Tan. On the power of symmetric LP and SDP relaxations. In *Proceedings of the 2014 IEEE 29th Conference on Computational Complexity*, pages 13–21. IEEE Computer Society, 2014. 22, 36, 44
- James R. Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proc. STOC*, pages 567–576, 2015. 22

- Richard Lipton and Ken Regan. Gödel’s lost letter and $p = np$. <https://rjlipton.wordpress.com/2012/11/29/barriers-to-ntp-proofs/>, 2012. 2
- Aleksandr N. Maksimenko. An analog of the cook theorem for polytopes. *Russian Mathematics*, 56(8):28–34, 2012. doi: 10.3103/S1066369X1208004X. Original Russian Text © A.N. Maksimenko, 2012, published in *Izvestiya Vysshikh Uchebnykh Zavedenii. Matematika*, 2012, No. 8, pp. 34–42. 23, 55, 58
- Kanstantsin Pashkovich. Tight lower bounds on the sizes of symmetric extensions of permutahedra and similar results. *Math. Oper. Res.*, 39(4):1330–1339, 2014. doi: 10.1287/moor.2014.0659. URL <http://dx.doi.org/10.1287/moor.2014.0659>. 21
- Thomas Rothvoß. The matching polytope has exponential extension complexity. *Proceedings of STOC*, pages 263–272, 2014. xiv, 19, 21, 22
- Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3:411–430, 1990. doi: 10.1137/0403036. 21
- Edward R. Swart. $P = NP$. Technical report, University of Guelph, 1986. 2
- Michael Trick. Michael trick’s operations research blog. <http://mat.tepper.cmu.edu/blog/?p=767>, 2009. 2
- Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Rev.*, 38: 49–95, 1996. 22
- Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs (extended abstract). In *Proc. STOC*, pages 223–228, 1988. xiv, 2, 20, 22, 44
- Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *J. Comput. System Sci.*, 43(3):441–466, 1991. ISSN 0022-0000. doi: 10.1016/0022-0000(91)90024-Y. xiv, 2, 20, 22, 44
- Günter M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1995. 29, 30