Face Lattice Computation under Symmetry

Face Lattice Computation under Symmetry

By

JONATHAN LI, B.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Applied Science

McMaster University

© Copyright by Jonathan Li, August 2008

MASTER OF APPLIED SCIENCE (2008)	McMaster University
(Computational Engineering and Science)	Hamilton, Ontario

TITLE:	Face Lattice Computation under Symmetry
AUTHOR:	Jonathan Li, B.Sc. (National Sun Yat-Sen University)
SUPERVISOR:	Dr. Antoine Deza
NUMBER OF PAGES:	x, 50.

Abstract

The last 15 years have seen a significant progress in the development of general purpose algorithms and software for polyhedral computation. Many polytopes of practical interest have enormous output complexity and are often highly degenerate, posing severe difficulties for known general purpose algorithms. They are, however, highly structured and attention has turned to exploiting this structure, particularly symmetry. We focus on polytopes arising from combinatorial optimization problems. In particular, we study the face lattice of the metric polytope associated with the well-known maxcut and multicommodity flow problems, as well as with finite metric spaces. Exploiting the high degree of symmetry, we provide the first complete orbitwise description of the higher layers of the face lattice of the metric polytope for any dimension. Further computational and combinatorial issues are presented.

Acknowledgments

The thesis was written under the guidance and with the help of my supervisor, Dr. Antoine Deza, whose valuable advices and extended knowledge helped me all along. My special thanks go to the members of the examination committee: Antoine Deza, Franya Franek (Chair) and Tamás Terlaky.

I appreciate the great aid and support from all the members of the Advanced Optimization Laboratory.

Finally, I am indebted to thank my family and their patience, understanding and continuous support.

Contents

List of Figures

Notations

$I\!R^d$	d dimensional Euclidean space
$dim(P)$	dimension of the smallest affine subspace containing polytope P
m	number of facets
K_n	complete graph on n nodes
c_n	cut polytope induced by K_n
m_n	metric polytope over n nodes
$\mathcal{F}(P)$	facet set of a polytope P
f_r^{d-t}	face of codimension t indexed by r
$\mathcal{F}(f_r^{d-t})$	facet set of a face f_r^{d-t}
$\widetilde{f}_s^{d-t}$	canonical representative of the orbit containing f_s^{d-t}
$\mathcal{L}^{d-t}$	set of all canonical representatives of the orbits of the faces of codimension t
I^{d-t}	number of orbits of codimension t faces; i.e. $I^{d-t} = \lvert\mathcal{L}^{d-t}\rvert$
L^{d-t}	$\{(\widetilde{f}_s^{d-t+1} \cap f_r^{d-1}), s = 1, ..., \lvert\mathcal{L}^{d-t}\rvert, r = 1, ..., \lvert\mathcal{F}(f_r^{d-t})\rvert\}$
$\mathrm{Is}(P)$	symmetry group of a polytope P

Chapter 1

Introduction

Convex polytopes are the d-dimensional analogues of 2-dimensional convex polygons and 3-dimensional convex polytopes. To a large extent the geometry of polytopes is just that of $\mathbb{R}^d$ itself. These geometric objects of relevant importance in various areas of mathematics and other disciplines have been studied since antiquity (e.g., the platonic solids). Interest in the theory of convex polytopes grew tremendously in the second half of the 20th century due to its relation with linear programming (i.e., optimizing a linear function over the solutions of a system of linear inequalities). DANTZIG's Simplex Algorithm, developed in the late 40's, showed that geometric and combinatorial knowledge of convex polytopes is key for finding and analyzing solution procedures for linear programming problems.

A convex polytope can be defined as the bounded intersection of a finite set $\mathcal{H}(\mathcal{P})$ of halfspaces. The well known theorem of Minkowski-Weyl states that polytopes can also be defined as the convex hull of its set $\mathcal{V}(\mathcal{P})$ of vertices. These two independent characterization of polytopes give rise to two closely related computational problems: how to compute $\mathcal{V}(\mathcal{P})$ from $\mathcal{H}(\mathcal{P})$, known as the vertex enumeration problem, and how to compute $\mathcal{H}(\mathcal{P})$ from $\mathcal{V}(\mathcal{P})$, known as the facet enumeration problem. These two problems are essentially equivalent under the point/hyperplane duality.

The vertex/facet enumeration of combinatorial polytopes, i.e. polytopes arising from combinatorial optimization problems, is often trivial for the very first cases and then suddenly the so-called combinatorial explosion occurs even for small instances. While these polytopes turn out to be quickly intractable for enumeration algorithms designed for general polytopes, algorithms using their rich combinatorial features can exhibit surprisingly strong performances. Recently, different research groups have proposed new enumeration techniques for combinatorial polytopes, in particular the metric polytope, that exploit their large symmetry groups making it possible tackle problems that until now have been intractable.

An even computationally harder problem is the face lattice enumeration. Previous works on the facial structure of the metric polytope include the orbitwise complete description of its face lattice in dimension 6 and 10, see [12], and of the top 3 layers of its face lattice for any dimension, see [11, 13]. In this thesis, we provide the orbitwise complete description of faces of codimension 4 for any n.

1.1 Preliminaries

1.1.1 Convex polytope

We recall some definitions and elementary properties concerning polyhedra. A complete presentation can be found in BAYER AND LEE [5], BRØNDSTED [8], GRÜNBAUM [19], MCMULLEN AND SHEPHARD [23], and ZIEGLER [27]. A *convex polyhedron* is an intersection of a finite number of closed half spaces in $\mathbb{R}^d$. Since we do not consider non-convex polyhedra, we often omit the term convex. A *polytope* is a bounded polyhedron.

Let P be a d-dimensional polytope, a linear inequality $c \cdot x \leq c_0$ is valid for P if it is satised for all points $x \in P$. A face f of P is any set of the form:

$$f = P \cap \{x \in \mathbb{R}^d : c \cdot x = c_0\}$$

where $c \cdot x \leq c_0$ is a valid inequality for P . The dimension of a face is the dimension of its affine hull. A *proper face* of P is a face f such that $f \neq \emptyset$. The faces of dimension 0, 1, d-2, and d-1 are respectively called the *vertices*, *edges*, *ridges*, and *facets* of the polytope. One of the earliest results in the field is the generalization by SCHLAFLI in 1852 of EULERS relation stating that the alternating sum of the number of i-faces (including the improper faces $\emptyset$ and P) equals zero. For the case $d = 3$, it was discovered by EULER in 1752. The *face lattice* of a convex polytope is the set of all its faces partially ordered by inclusion. Two polytopes are *combinatorially equivalent*, respectively *dual*, if there is a bijection between their faces which preserves, respectively reverses, the inclusion relation.

Figure 1.1: Platonic solids

A d-dimensional polytope with exactly d+1 vertices is called a *simplex*. A polytope such that each vertex belongs to exactly d edges is *simple*, and a polytope such that each facet contains exactly d vertices is *simplicial.* A d-dimensional polytope is called k-simplicial if each k-face is a simplex. The dual of a k-simplicial polytope is called k-simple. Figure 1.1 illustrates different types of polytopes in dimension 3, namely the five platonic solids: tetrahedron, cube, octahedron, dodecahedron and isocahedron. We recall some denitions and elementary properties concerning the graph of a polytope. A complete presentation can be found in BRØNDSTED [8] and ZIEGLER [27]. The main reference for the general graph theory is BROUWER, COHEN AND NEUMAIER [9]. The vertices and edges of a d-dimensional polytope P clearly form an undirected graph $G(P)$ called the *skeleton* of P. The *diameter* $\delta(P)$ of a polytope P is the diameter of its skeleton, that is, the smallest number k such that any two vertices of P can be connected by a path with at most k edges. In this thesis we will consider only non-redundant, full dimensional, and bounded convex polyhedra.

1.2 Face lattice enumeration

1.2.1 Vertex enumeration

Given a polytope P defined by the linear inequalities associated with the set $\mathcal{F}(\mathcal{P})$ of its facets, the computation of its vertex set $\mathcal{V}(\mathcal{P})$ is referred to as the vertex enumeration problem. The main vertex enumeration algorithms can be viewed as not just generating all vertices of a polytope P, but actually generating the skeleton of P, i.e. the graph formed by its vertices and edges. There are essentially two main classes of algorithms for producing these graphs: graph traversal algorithms and incremental algorithms.

A graph traversal algorithm, also called a pivoting method, first finds one vertex of P and then identifies all vertices (and edges) by moving from one vertex to an adjacent one. In this method, each vertex v is described by a basis - i.e. d affinely independent inequalities containing v. Moving from one vertex to an adjacent one amounts to changing one member of the basis in some proper way. The basic incremental algorithm first selects $d + 1$ affinely independent inequalities and computes the vertices and edges of the associated d-simplex. Then at each step k one of the remaining inequalities H_k is inserted and the vertex and edge description is updated by removing the vertices cut off by the newly inserted inequality H_k and adding new vertices (and edges) created by the intersections of edges of the intermediate polytope P_k with the newly inserted inequality H_k.

For a detailed presentation of the main existing algorithms we refer to Avis, Bremmer and Seidel [1] and references therein. Even though these algorithms often perform quite well for many cases, in particular for low dimensional and simple polytopes, and despite the fact that the vertex enumeration problem has been extensively studied by many authors see for instance [2, 10, 17, 22, 24], there is no satisfying algorithm for generating the vertices of a general polytope given by its facets.

1.2.2 Face lattice enumeration

For most of the combinatorial polytopes, the number of faces usually grows extremely large as the dimension of the faces is getting close to roughly half the dimension of the polytope: Face lattices are usually "fat" making the computation of the full face lattice of a polytope extremely hard.

In general, a proper face f_r^{d-t} of P can be defined either by the subset

$\mathcal{F}(f_r^{d-t})$ of facets containing f_r^{d-t} or as the convex hull of the vertices $\mathcal{V}(f_r^{d-t})$ belonging to f_r^{d-t}. The codimension of a $(d-t)$-face f_r^{d-t} is t. Given the facet set $\mathcal{F}(P)$, the face enumeration problem consists in enumerating all the faces of P in terms of $\mathcal{F}(f_r^{d-t})$. A face enumeration algorithm usually first generates the set L^{d-t} of all the possible intersections between facets and the face of codimension $t-1$, removes the duplicates, and then determines the facet set and computes the rank of the remaining intersections. The computation can quickly become intractable, when the number of intersections becomes too large. To exploit the symmetries displayed by most combinatorial polytopes, an orbitwise face enumeration algorithm is proposed in Chapter 3.

1.3 Combinatorial polytope

Combinatorial polytopes, i.e. polytopes arising from combinatorial optimization problems, are usually associated with the complete directed graph D_n or the complete undirected graph K_n on n nodes. Solving an instance of a combinatorial optimization problem means finding a feasible solution of minimum or maximum cost. The combinatorial polytope is the convex hull of the set of vectors representing the feasible solutions. A standard approach is to try to describe the polytope in terms of linear inequalities with the hope of applying the tools of linear programming. For instance, for matchings, spanning trees, and several other structures, we are able to get a compact description of the convex hull in terms of linear inequalities. In many cases, though, it is hard to obtain such a description. Combinatorial polytopes become quickly intractable for enumeration algorithms designed for solving general polytopes.

1.4 Cut polytope

The cut polytope c_n is the convex hull of the incidence vectors of all the cuts of K_n. More precisely, given a subset S of $\{1...n\}$, the cut determined by S consists of the pairs (i,j) of elements of $\{1...n\}$ such that exactly one of i,j is in S. By $\delta(S)$ we denote both the cut and its incidence vector in $I\!R^n$; that is, $\delta(S)_{ij} = 1$ if exactly one of i,j is in S and 0 otherwise for $1 \leq i \leq j \leq n$. So, $\delta(S)_{ij}$ could be considered as coordinates of a point in $I\!R^n$. The cut polytope is the convex hull of all 2^{n-1} cuts, and the cut cone is the conic hull of all $2^{n-1}-1$ nonzero cuts.

One of the well-known applications for the cut polytope is the maxcut problem. The maxcut problem could be stated as follows: given a graph $G = (N, E)$ and nonnegative weights $w_e, e \in E$, assigned to its edges, the maxcut problem consists in finding a cut $\delta(S)$ whose weight $\sum_{e\in\delta(S)} w_e$ is as large as possible. It is well known that it is an *NP*-complete problem. By setting $w_e = 0$ if e is not an edge of G, we can consider without loss of generality the complete graph K_n. Then the maxcut problem can be stated as a linear programming problem over the cut polytope c_n as follows:

$$\max w^T \cdot x$$

$$\text{such that } x \in c_n$$

1.5 Metric polytope

The metric polytope m_n is one of well-studied relaxations of the cut polytope. It also can be defined in terms of a finite metric space in the following way. For all 3-sets $\{i, j, k\} \in \{1, ..., n\}$, we consider the inequalities:

$$x_{ij} - x_{ik} - x_{jk} \leq 0 \qquad (1)$$

$$x_{ij} + x_{ik} + x_{jk} \leq 2 \qquad (2)$$

(1) induces $3\binom{n}{3}$ facets, which define the metric cone. Then, bounding the latter by the $\binom{n}{3}$ facets induced by (2), we obtain the metric polytope. While the cut cone is the conic hull of all, up to a constant multiple, $\{0, 1\}$-valued extreme rays of the metric cone, the cut polytope is the convex hull of all $\{0, 1\}$-valued vertices of the metric polytope.

We have $c_n \subseteq m_n$ with equality only for $n \leq 4$. Any facet of the metric polytope contains a facet of the cut polytope and the vertices of the cut polytope are vertices of the metric polytope, in fact the cuts are precisely the integral vertices of the metric polytope. Actually the metric polytope m_n wraps the cut polytope c_n very tightly since, in addition to the vertices, all edges and 2-faces of c_n are also faces of m_n, for 3-faces it is false for $n \geq 4$. Any two cuts are adjacent both on c_n and on m_n. Since the metric polytope is a relaxation of the cut polytope, optimizing $w^T \cdot x$ in the previous section over m_n instead of c_n provides an upper bound for the maxcut problem.

1.5.1 Faces of the metric polytope

The metric polytope m_n is a $\binom{n}{2}$-polytope with $4\binom{n}{3}$ facets inscribed in the cube $[0, 1]^{\binom{n}{2}}$. We recall some results on the vertices of the metric polytope. The cuts are the only integral vertices of m_n. All other vertices with are not fully fractional are so-called trivial-extensions of a vertex of m_{n-1}. In other words, the new vertices are the fully fractional ones. The $(\frac{1}{3}, \frac{2}{3})$-valued fully fractional vertices are well studied and include the anticut orbit formed by the 2^{n-1} anticuts $\bar{\delta}(S) = \frac{2}{3}(1, ..., 1) - \frac{1}{3}\delta(S)$, where $\delta(S)$ represents both the cut and its incidence vector in $\mathbb{R}^{\binom{n}{2}}$. Consider the mapping: ϕ_0: $R^{\binom{n-1}{2}} \Rightarrow R^{\binom{n}{2}}$, defined by $\phi_0(v)_{ij} = v_{ij}$ for $1 \leq i < j \leq n-1$, $\phi_0(v)_{i,n} = v_{1,i}$ for $2 \leq i \leq n-1$ and $\phi_0(v)_{1,n} = 0$, both $\phi_0(v)$ and its switching by $\delta(\{n\})$ are called trivial extensions of v.

While the diameter of the dual of the metric polytope is 2, the diameters of the dual cut polytope and m_n are respectively conjectured to be 4 and 3, see [11, 21]. We recall two independent conjectures concerning the combinatorial structure of the metric polytope: the dominant clique conjecture [21] stating that the cut vertices form a dominating set, and the non-cut set conjecture [15] stating that for $n \geq 6$, the restriction of the skeleton to the non-cut vertices is connected while the dominant clique conjecture was disproved in [15], the non-cut set conjecture is still open. The full face-lattice enumeration has been performed for m_4 and m_5, see [12]. The orbitwise descriptions of the faces of m_n of codimension 1, 2 and 3 was given in [11, 13].

1.6 Symmetry group and orbits

In this thesis we consider polytopes associated with problems that are symmetric. We recall that the *symmetry group* Is(P) of a polytope P is the group of isometries preserving P. Typical examples of polytopes with large symmetry group are polytopes associated with problems arising from the complete directed graph D_n or the complete undirected graph K_n on n nodes. Some such well-known polytopes are: the *traveling salesman polytope* tsp_n which is the convex hull of all the incidence vectors of all Hamiltonian cycles of K_n and the *linear ordering polytope* lo_n which is the convex hull of the incidence vectors of all acyclic tournaments of D_n. The isometries preserving tsp_n are induced by the $n!$ permutations on $\{1, 2, ..., n\}$, that is, $\mathrm{Is}(tsp_n) \simeq Sym(n)$. In this thesis, we consider polytopes with even larger symmetry group: the cut and metric polytope. More precisely, for $n \geq 5$, $\mathrm{Is}(m_n) = \mathrm{Is}(c_n)$ is induced by the $n!$ permutations on $\{1, 2, ..., n\}$ and the 2^{n-1} switching reflections by cuts and we have $|\mathrm{Is}(m_n)| = 2^{n-1}n!$, see [14]. As these symmetries preserve the adjacency relations and the linear independency, all faces of m_n are partitioned into orbits of faces equivalent under permutations and switchings.

Chapter 2

Face Lattice Computation Under Symmetry

As the face lattice enumeration over combinatorial polytopes turns out to be intractable due to the exponentially growing combinatorial structure, we focus on enumerating the face lattice by exploiting symmetry. We consider the problem of enumerating the orbits of the face lattice with a given facet set $\mathcal{F}(P) = \{f_1^{d-1}, f_2^{d-1}, ..., f_m^{d-1}\}$. By duality, the methods we discuss here also apply when the polytope is defined as the convex hull of its vertex set.

2.1 Decomposition method

The decomposition method consists in enumerating the orbits of the face lattice by decomposing the original problem into several subproblems. Subproblems here refer to the smaller input size regarding current available vertex (resp. face) enumeration algorithm, and they are usually defined with respect to certain orbit sets. For example, in the *incidence decomposition method* (resp. *adjacency decomposition method*), the subproblems are corresponding to the orbits of facets (resp. the orbits of vertices), while in the orbitwise face enumeration algorithm, the subproblems are corresponding to the orbits of faces. Although different variants of decomposition method require different analysis of the computation, we outline here the common tasks this type of method usually involves:

(i) decomposing the original problem into subproblems with respect to certain sets of orbits,

(ii) applying traditional vertex (resp. face) enumeration algorithm for each subproblem,

(iii) identifying the canonical representative for the vertex (resp. face) set result from *(ii)*,

(iv) updating the orbit list until the orbit list is completed.

In *(i)*, by taking advantage of the complete description of faces up to symmetry, the computation of the decomposed subproblems are guaranteed to generate all the orbits of desirable vertex (resp. face) set. For most of the orbitwise enumeration algorithms, the efficiency of the algorithm results from the trade-off among *(i)*, *(ii)*, *(iii)*. In many cases, the performances are empirical and rely on heuristics such as skipping the high degeneracy in the adjacency decomposition method.

2.2 Incidence decomposition method

The incidence decomposition method reduces the problem of vertex enumeration into a number of smaller subproblems with respect to orbits of facets, in which the algorithm generates vertices that are incident to the chosen facets.

Let P be a polytope in $I\!R^d$ generated by facet set $\mathcal{F}(P)$ $=\{f_1^{d-1},...,f_m^{d-1}\}$, the set $\mathcal{F}(P) = \{f_1^{d-1},...,f_m^{d-1}\}$ is partitioned into orbits under the action of $\mathrm{Is}(P)$. The algorithm generates a list of $\mathrm{Is}(P)$-inequivalent vertices of P incident to each canonical representative for each orbit of facets. Then, all the computed vertices are merged to a list of $\mathrm{Is}(P)$-inequivalent vertices of P. The certificate of the complete enumeration of vertices up to symmetry comes from the fact that every $\mathrm{Is}(P)$-orbit of vertices of P contains a vertex which is incident to one of the chosen canonical representatives of facets. In other words, it is sufficient to enumerate all the orbits of vertices by enumerating the vertices of each canonical representative of facets.

To compute the vertices incident to a given facet f_r^{d-1}, the method enumerates the vertices of lower dimensional polytopes. For each subproblem we may not have to consider all the facets of P because some of the facets may not be incident to the vertices incident to the canonical representative $\tilde{f}_r^{d-1}$. The facets not incident to $\tilde{f}_r^{d-1}$ correspond to redundant inequalities for $P_r^* := \{P \cap f_r^{d-1}\}$. The lower the number of incident facets of each canonical representative, the more computational gain we could have by applying general vertices enumeration algorithm for each lower-dimensional polytope. If the computational cost is still too high for each subproblem, the method could be applied recursively.

2.3 Adjacency decomposition method

The adjacency decomposition method traverses the orbits of vertices directly. Starting from a (set of) initial $\mathrm{Is}(P)-$inequivalent vertex (vertices), it traverses the adjacency graph of vertex orbits. To traverse adjacent vertices, the method can either apply incremental algorithm or graph traversal algorithm. When applying graph traversal algorithm, the method traverses the so-called basis graph. The nodes of the basis graph are the bases, and the edges are the pairs of adjacent bases.

2.4 Orbitwise face enumeration

The *orbitwise face enumeration* generates the set L^{d-t} of all intersections between the canonical representatives $\mathcal{L}^{d-t+1}$ and facets from $\mathcal{F}(P)$, and extracts $\mathcal{L}^{d-t}$ from L^{d-t} by applying orbitwise equivalency checks.

Similarly to the incidence decomposition method that lowers the number of possible orbits of vertices to enumerate by only enumerating those being incident to the orbits of facets, the generation of faces of codimension t intersections can be obtained by intersecting the orbits of faces of codimension $(t-1)$ with all facets. After generating the list of possible orbits of faces of codimension t up to symmetry, the algorithm further computes the canonical representatives using the list. Further details can be found in the following chapters, also see [13].

2.5 Refinement using symmetry

2.5.1 Recursion

The incidence decomposition method and the adjacency decomposition method reduce the vertex enumeration problem over a $(d-t)-$dimensional polytope P to a number of vertex enumeration subproblems over polytopes in $(d-t-1)$ dimension. These lower dimensional subproblems may still be difficult to solve for general enumeration algorithms. If so, we might apply the methods recursively to further lower the input size. In both methods, after reaching the step computing a list of vertices for a $(d-t)$ dimensional polytope f_r^{d-t}, the further exploitation of symmetry with respect to f_r^{d-t} could be done; that is computing $\mathrm{Is}(f_r^{d-t})$ where $\mathrm{Is}(f_r^{d-t})$ is some symmetry group acting on the face lattice of f_r^{d-t}. With $\mathrm{Is}(f_r^{d-t})$, we can then obtain a list of $\mathrm{Is}(f_r^{d-t})$-inequivalent vertices of f_r^{d-t}. In a post processing step, we then have to obtain a list of $\mathrm{Is}(P)$-inequivalent vertices out of the set of $\mathrm{Is}(f_r^{d-t})$-inequivalent vertices of all subproblems, $r = 1, ..., I^{d-t}$.

2.5.2 Adjacency decomposition pruning method

Adjacency decomposition pruning method is the refinement of the adjacency decomposition method. Consider vertices v_0 and v_1 that are equivalent under some symmetry of the basis automorphism group. This same symmetry acts as an isomorphism between corresponding basis graphs. In other words, the neighborhood of v_0 is symmetric to the neighborhood of v_1. It follows that when we discover a basis (vertex) B defining a new orbit, the orbit shall be visited only once, since the combinatorial structures of the vertices in the same orbit are equivalent. In other words, it is suffcient to enumerate all the orbits of vertices by exploring the neighborhood of canonical representatives. Although this pruning method does not reduce the number of orbits of bases explored, it can reduce the number of actual bases visited and save the computational cost result from computing the adjacent vertices.

2.5.3 A pivoting method using symmetry

Pivoting methods are among most successful methods for vertex enumeration problem and it is natural to consider whether pivoting technique can be adapted to the symmetric setting. In the typical case, generating the entire basis graph is impractical due to the large number of bases that correspond to each facet in the degenerate case. The performance of pivoting method under symmetry is determined by the number of orbits of bases with respect to the *basis automorphism group* defined as the subgroup of the combinatorial automorphism group that acts on the basis graph; for further details, see [7]

Chapter 3

Orbitwise Face Enumeration Algorithm

This section presents the complete characterization of the orbits of faces of codimension 4 for the metric polytope m_n for $n \geq 3$.

3.1 Orbitwise enumeration algorithm

Given a polytope P defined by its (non-redundant) facet set $\mathcal{F}(P) = \{f_1^{d-1},\ldots, f_m^{d-1}\}$. The algorithm first computes the list $\mathcal{L}^{d-1} = \{\tilde{f}_1^{d-1}, \ldots, \tilde{f}_{I^{d-1}}^{d-1}\}$ of all the canonical representatives of the orbits of facets. Then the algorithm generates the set $L^{d-2} = \{\tilde{f}_s^{d-1} \cap f_r^{d-1} : s = 1, \ldots, I^{d-1}, r = 1, \ldots, m\}$. To identify and keep only the $(d-2)$-faces, the algorithm computes the dimension of each subface $\tilde{f}_s^{d-1} \cap f_r^{d-1}$. Then it computes the list of canonical representatives of orbits of $(d-2)$-faces $\mathcal{L}^{d-2} = \{\tilde{f}_1^{d-2}, \ldots, \tilde{f}_{I^{d-2}}^{d-2}\}$ from the L^{d-2}. In general, the algorithm computes $\mathcal{L}^{d-t}$ from $\mathcal{L}^{d-t+1}$ through the following steps:

1. generating the set L^{d-t} by intersecting canonical representatives $\tilde{f}_s^{d-t+1}$ with facets f_r^{d-1} for $s = 1, \ldots, I^{d-t+1}$ and $r = 1, \ldots, m$,

2. computing the set $\mathcal{F}(\tilde{f}^{d-t+1} \cap f^{d-1})$ of all facets containing $\tilde{f}^{d-t+1} \cap f^{d-1}$ and then its rank $\dim(\tilde{f}^{d-t+1} \cap f^{d-1})$

3. for $\dim(\tilde{f}^{d-t+1} \cap f^{d-1}) = d - t$, computing the canonical representative $\tilde{f}^{d-t}$ of $\tilde{f}^{d-t+1} \cap f^{d-1}$

The algorithm terminates after the list $\mathcal{L}^0$ of canonical representatives of the orbits of vertices is computed. The algorithm performs better when the symmetry group $\mathrm{Is}(P)$ is larger since the number of orbits could be relatively small. One of major computational costs arises from the computation of the canonical representative $\tilde{f}_r^{d-t}$ of the orbit $O_{f_r^{d-t}}$ generated by a face f_r^{d-t} and the orbitwise equivalency check. The computation of $\dim(\tilde{f}^{d-t+1} \cap f^{d-1})$ is performed by computing the rank of $\mathcal{F}(\tilde{f}^{d-t+1} \cap f^{d-1})$. Assuming $\tilde{f_s}^{d-t+1} = \{f_1^{d-1}, ..., f_{t-1}^{d-1}\}$ and $f_r^{d-1} = f_t^{d-1}$ by re-ordering the given facets and defining $f_i^{d-1} = \{a_i^T x = b_i, x \in P\}$, the following LP_j, $j \in \{t+1, ..., m\}$, is used to determine $\mathcal{F}(\tilde{f_s}^{d-t+1} \cap f_r^{d-1})$:

$$\begin{array}{lll} \max & a_j^T x - b_j & \\ \text{s.t.} & a_i^T x - b_i = 0 & i = 1, ..., t \\ & a_i^T x - b_i \geq 0 & i = t+1, ..., m \\ & x \in I\!R^{\binom{n}{2}} & \end{array}$$

The LP_j checks if the facet $f_{j \geq t+1}^{d-1}$ contains the intersection $\tilde{f}_s^{d-t+1} \cap f_j^{d-1}$ by trying to push away $f_{j \geq t+1}^{d-1}$ from $\tilde{f_s}^{d-t+1} \cap f_r^{d-1}$. If $a_j^T x - b_j > 0$, $f_{j \geq t+1}^{d-1}$ does not contain the intersection and so it can be "detached" from the intersection. On the other hand, $f_{j \geq t+1}^{d-1}$ does contain the intersection if $a_j^T x - b_j = 0$.

3.2 Previous results and computation

3.2.1 Full face lattice of the metric polytope m_5

Using the algorithm presented at section 3.1, the full face lattice for m_5 is presented.

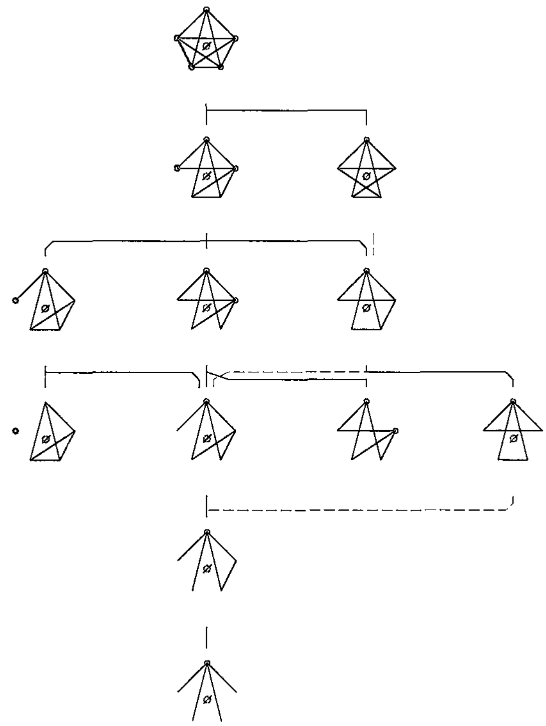

Figure 3.1: Non-simplices of the face lattice of m_5

3.2.2 Orbitwise description of face of codimension 2 and 3 of the metric polytope for any n

In the following the orbitwise description of face of codimension 2 and codimension 3 for any n are presented, for details see [13].

Orbit $O_{f_i^2}$	Representative f_i^2	m_n for which f_i^2 is a $(d-2)$-face	$\lvert O_{f_i^2}\rvert$
$O_{f_1^2}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4}$	$m_{n\geq 4}$	$160\binom{n}{4}$
$O_{f_2^2}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5}$	$m_{n\geq 5}$	$48\binom{n}{5}$
$O_{f_3^2}$	$\Delta_{1,2,3} \cap \Delta_{4,5,6}$	$m_{n\geq 6}$	$240\binom{n}{6}$

Orbitwise description of face of codimension 2

Orbit $O_{f_i^3}$	Representative f_i^3	m_n for which f_i^3 is a $(d-3)$-face	$\lvert O_{f_i^3}\rvert$
$O_{f_1^3}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,\overline{3},4}$	$m_{n\geq 4}$	$32\binom{n}{4}$
$O_{f_2^3}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,4}$	$m_{n\geq 4}$	$24\binom{n}{4}$
$O_{f_3^3}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{2,3,\overline{4}}$	$m_{n\geq 5}$	$160\binom{n}{5}$
$O_{f_4^3}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,5}$	$m_{n\geq 5}$	$960\binom{n}{5}$
$O_{f_5^3}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5}$	$m_{n\geq 5}$	$480\binom{n}{5}$
$O_{f_6^3}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{2,3,4}$	$m_{n\geq 5}$	$480\binom{n}{5}$
$O_{f_7^3}$	$\Delta_{1,2,3} \cap \Delta_{1,3,5} \cap \Delta_{2,4,6}$	$m_{n\geq 6}$	$5760\binom{n}{6}$
$O_{f_8^3}$	$\Delta_{1,2,3} \cap \Delta_{1,2,6} \cap \Delta_{1,4,5}$	$m_{n\geq 6}$	$5760\binom{n}{6}$
$O_{f_9^3}$	$\Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{\overline{2},3,6}$	$m_{n\geq 6}$	$3840\binom{n}{6}$
$O_{f_{10}^3}$	$\Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{2,3,6}$	$m_{n\geq 6}$	$3840\binom{n}{6}$
$O_{f_{11}^3}$	$\Delta_{1,2,5} \cap \Delta_{1,2,7} \cap \Delta_{3,4,6}$	$m_{n\geq 7}$	$6720\binom{n}{7}$
$O_{f_{12}^3}$	$\Delta_{1,2,7} \cap \Delta_{1,3,5} \cap \Delta_{2,4,6}$	$m_{n\geq 7}$	$6720\binom{n}{7}$
$O_{f_{13}^3}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{1,6,7}$	$m_{n\geq 7}$	$40320\binom{n}{7}$
$O_{f_{14}^3}$	$\Delta_{1,3,5} \cap \Delta_{1,7,8} \cap \Delta_{2,4,6}$	$m_{n\geq 8}$	$53760\binom{n}{8}$
$O_{f_{15}^3}$	$\Delta_{1,3,5} \cap \Delta_{2,4,6} \cap \Delta_{7,8,9}$	$m_{n\geq 9}$	$17920\binom{n}{9}$
$O_{f_{16}^3}$	$\Delta_{1,2,3} \cap \Delta_{1,2,\overline{3}} \cap \Delta_{1,2,4}$	m_4	$2\binom{n}{2}$

Orbitwise description of face of codimension 3

3.3 The faces of codimension 4 of the metric polytope for any n

As mentioned in Section 3.2, the first 3 upper layers of m_n are known for any n. We have $I^{d-1}(m_{n\geq 3})$=1, $I^{d-2}(m_{n\geq 6})$=3, $I^{d-3}(m_{n\geq 9})$=15 and by Theorem **3.3.1**, we get $I^{d-4}(m_{n\geq 12})$=94.

Theorem 3.3.1 *For $n \geq 12$, the face of codimension 4 of the metric polytope m_n are partitioned into 94 orbits equivalent under permutations and switchings. For $n = 4, ..., 11$ the face of codimension 4 are partitioned into 2, 10, 34, 61, 79, 88, 92, 93 orbits respectively.*

Orbit $O_{f_i^4}$	Representative f_i^4	m_n for which f_i^4 is a $(d-4)$-face
$O_{f_1^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,4} \cap \Delta_{2,3,4}$	$m_{n \geq 4}$
$O_{f_2^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{1,3,4}$	$m_{n \geq 5}$
$O_{f_3^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{1,4,5}$	$m_{n \geq 5}$
$O_{f_4^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{1,\overline{3},4}$	$m_{n \geq 5}$
$O_{f_5^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{\overline{3},4,5}$	$m_{n \geq 5}$
$O_{f_6^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,\overline{5}} \cap \Delta_{1,3,4} \cap \Delta_{\overline{2},4,5}$	$m_{n \geq 5}$
$O_{f_7^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{3,4,5}$	$m_{n \geq 5}$
$O_{f_8^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,4} \cap \Delta_{2,3,5}$	$m_{n \geq 5}$
$O_{f_9^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,5} \cap \Delta_{1,3,4} \cap \Delta_{2,4,5}$	$m_{n \geq 5}$
$O_{f_{10}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,5} \cap \Delta_{1,4,5} \cap \Delta_{\overline{2},4,6}$	$m_{n \geq 6}$
$O_{f_{11}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,5} \cap \Delta_{1,4,6} \cap \Delta_{\overline{2},4,5}$	$m_{n \geq 6}$
$O_{f_{12}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{\overline{2},4,6} \cap \Delta_{3,\overline{5},6}$	$m_{n \geq 6}$
$O_{f_{13}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,\overline{3},4} \cap \Delta_{1,5,6}$	$m_{n \geq 6}$
$O_{f_{14}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{\overline{2},4,6} \cap \Delta_{3,5,6}$	$m_{n \geq 6}$
$O_{f_{15}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{2,3,6}$	$m_{n \geq 6}$
$O_{f_{16}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,4} \cap \Delta_{1,5,6}$	$m_{n \geq 6}$
$O_{f_{17}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,4} \cap \Delta_{2,5,6}$	$m_{n \geq 6}$
$O_{f_{18}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,5} \cap \Delta_{1,4,5} \cap \Delta_{2,4,6}$	$m_{n \geq 6}$
$O_{f_{19}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,6} \cap \Delta_{1,4,5} \cap \Delta_{3,4,5}$	$m_{n \geq 6}$
$O_{f_{20}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,\overline{4}} \cap \Delta_{1,2,5} \cap \Delta_{3,4,6}$	$m_{n \geq 6}$
$O_{f_{21}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,2,6} \cap \Delta_{1,3,4} \cap \Delta_{\overline{2},3,4}$	$m_{n \geq 6}$
$O_{f_{22}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{1,3,6}$	$m_{n \geq 6}$
$O_{f_{23}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{3,4,6}$	$m_{n \geq 6}$
$O_{f_{24}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,5} \cap \Delta_{1,3,4} \cap \Delta_{1,4,6}$	$m_{n \geq 6}$
$O_{f_{25}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{2,4,6}$	$m_{n \geq 6}$
$O_{f_{26}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{3,5,6} \cap \Delta_{4,5,6}$	$m_{n \geq 6}$
$O_{f_{27}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{1,3,4} \cap \Delta_{2,3,6}$	$m_{n \geq 6}$
$O_{f_{28}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,2,6} \cap \Delta_{1,3,4} \cap \Delta_{2,3,4}$	$m_{n \geq 6}$
$O_{f_{29}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{1,3,4} \cap \Delta_{\overline{2},3,6}$	$m_{n \geq 6}$
$O_{f_{30}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,6} \cap \Delta_{\overline{1},4,5} \cap \Delta_{3,4,5}$	$m_{n \geq 6}$
$O_{f_{31}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,\overline{5}} \cap \Delta_{1,4,6} \cap \Delta_{\overline{2},4,5}$	$m_{n \geq 6}$
$O_{f_{32}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{1,2,6}$	$m_{n \geq 6}$
$O_{f_{33}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,5} \cap \Delta_{1,4,6} \cap \Delta_{2,4,5}$	$m_{n \geq 6}$
$O_{f_{34}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{3,5,6} \cap \Delta_{\overline{4},5,6}$	$m_{n \geq 6}$
$O_{f_{35}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,2,7} \cap \Delta_{1,3,5} \cap \Delta_{\overline{2},3,6}$	$m_{n \geq 7}$

Orbit $O_{f_i^4}$	Representative f_i^4	m_n for which f_i^4 is a $(d-4)$-face
$O_{f_{36}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,\overline{4}} \cap \Delta_{1,5,6} \cap \Delta_{2,4,7}$	$m_{n\geq 7}$
$O_{f_{37}^4}$	$\Delta_{1,2,5} \cap \Delta_{\overline{1},3,4} \cap \Delta_{1,6,7} \cap \Delta_{2,3,4}$	$m_{n\geq 7}$
$O_{f_{38}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{\overline{3},4,5} \cap \Delta_{5,6,7}$	$m_{n\geq 7}$
$O_{f_{39}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,\overline{3},4} \cap \Delta_{5,6,7}$	$m_{n\geq 7}$
$O_{f_{40}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,6} \cap \Delta_{2,5,7} \cap \Delta_{3,\overline{4},5}$	$m_{n\geq 7}$
$O_{f_{41}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,4} \cap \Delta_{5,6,7}$	$m_{n\geq 7}$
$O_{f_{42}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{1,6,7}$	$m_{n\geq 7}$
$O_{f_{43}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{2,6,7}$	$m_{n\geq 7}$
$O_{f_{44}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,4} \cap \Delta_{1,4,5} \cap \Delta_{2,6,7}$	$m_{n\geq 7}$
$O_{f_{45}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,3,6} \cap \Delta_{1,4,7} \cap \Delta_{\overline{2},3,4}$	$m_{n\geq 7}$
$O_{f_{46}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{3,6,7}$	$m_{n\geq 7}$
$O_{f_{47}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,7} \cap \Delta_{1,5,6}$	$m_{n\geq 7}$
$O_{f_{48}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,6} \cap \Delta_{2,5,7} \cap \Delta_{\overline{3},4,5}$	$m_{n\geq 7}$
$O_{f_{49}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,2,7} \cap \Delta_{1,3,5} \cap \Delta_{2,3,6}$	$m_{n\geq 7}$
$O_{f_{50}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,4} \cap \Delta_{1,5,6} \cap \Delta_{2,4,7}$	$m_{n\geq 7}$
$O_{f_{51}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,3,4} \cap \Delta_{1,6,7} \cap \Delta_{2,3,4}$	$m_{n\geq 7}$
$O_{f_{52}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{3,4,5} \cap \Delta_{5,6,7}$	$m_{n\geq 7}$
$O_{f_{53}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,7} \cap \Delta_{1,4,5} \cap \Delta_{2,4,6}$	$m_{n\geq 7}$
$O_{f_{54}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,3,7} \cap \Delta_{1,4,5} \cap \Delta_{\overline{2},4,6}$	$m_{n\geq 7}$
$O_{f_{55}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,2,6} \cap \Delta_{1,3,4} \cap \Delta_{3,4,7}$	$m_{n\geq 7}$
$O_{f_{56}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{1,3,6} \cap \Delta_{1,3,7}$	$m_{n\geq 7}$
$O_{f_{57}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,6} \cap \Delta_{3,4,5} \cap \Delta_{4,5,7}$	$m_{n\geq 7}$
$O_{f_{58}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,3,6} \cap \Delta_{1,4,7} \cap \Delta_{2,3,4}$	$m_{n\geq 7}$
$O_{f_{59}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,6} \cap \Delta_{2,5,7} \cap \Delta_{3,4,5}$	$m_{n\geq 7}$
$O_{f_{60}^4}$	$\Delta_{1,2,6} \cap \Delta_{1,3,4} \cap \Delta_{2,5,7} \cap \Delta_{3,4,5}$	$m_{n\geq 7}$
$O_{f_{61}^4}$	$\Delta_{1,2,6} \cap \Delta_{1,3,4} \cap \Delta_{2,5,7} \cap \Delta_{3,4,\overline{5}}$	$m_{n\geq 7}$
$O_{f_{62}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{2,3,\overline{4}} \cap \Delta_{6,7,8}$	$m_{n\geq 8}$
$O_{f_{63}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,3,6} \cap \Delta_{2,4,7} \cap \Delta_{\overline{3},4,8}$	$m_{n\geq 8}$
$O_{f_{64}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{6,7,8}$	$m_{n\geq 8}$
$O_{f_{65}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,5,6} \cap \Delta_{3,7,8}$	$m_{n\geq 8}$
$O_{f_{66}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,8} \cap \Delta_{1,4,5} \cap \Delta_{1,6,7}$	$m_{n\geq 8}$
$O_{f_{67}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,8} \cap \Delta_{1,4,5} \cap \Delta_{2,6,7}$	$m_{n\geq 8}$
$O_{f_{68}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{1,3,8} \cap \Delta_{2,6,7}$	$m_{n\geq 8}$
$O_{f_{69}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{1,6,7} \cap \Delta_{2,3,8}$	$m_{n\geq 8}$
$O_{f_{70}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,5,6} \cap \Delta_{2,3,4} \cap \Delta_{4,7,8}$	$m_{n\geq 8}$

Orbit $O_{f_i^4}$	Representative f_i^4	m_n for which f_i^4 is a $(d-4)$-face
$O_{f_{71}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,6,7} \cap \Delta_{2,3,4} \cap \Delta_{3,4,8}$	$m_{n\geq 8}$
$O_{f_{72}^4}$	$\Delta_{1,2,4} \cap \Delta_{\overline{1},3,8} \cap \Delta_{1,5,6} \cap \Delta_{2,3,7}$	$m_{n\geq 8}$
$O_{f_{73}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,3,6} \cap \Delta_{\overline{2},3,4} \cap \Delta_{4,7,8}$	$m_{n\geq 8}$
$O_{f_{74}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,8} \cap \Delta_{5,6,7}$	$m_{n\geq 8}$
$O_{f_{75}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{2,3,4} \cap \Delta_{6,7,8}$	$m_{n\geq 8}$
$O_{f_{76}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,2,6} \cap \Delta_{3,4,7} \cap \Delta_{3,4,8}$	$m_{n\geq 8}$
$O_{f_{77}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,3,8} \cap \Delta_{1,5,6} \cap \Delta_{2,3,7}$	$m_{n\geq 8}$
$O_{f_{78}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,3,6} \cap \Delta_{2,3,4} \cap \Delta_{4,7,8}$	$m_{n\geq 8}$
$O_{f_{79}^4}$	$\Delta_{1,2,5} \cap \Delta_{1,3,6} \cap \Delta_{2,4,7} \cap \Delta_{3,4,8}$	$m_{n\geq 8}$
$O_{f_{80}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{3,5,6} \cap \Delta_{7,8,9}$	$m_{n\geq 9}$
$O_{f_{81}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,9} \cap \Delta_{1,4,5} \cap \Delta_{6,7,8}$	$m_{n\geq 9}$
$O_{f_{82}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,2,5} \cap \Delta_{3,6,7} \cap \Delta_{3,8,9}$	$m_{n\geq 9}$
$O_{f_{83}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{\overline{2},3,9} \cap \Delta_{6,7,8}$	$m_{n\geq 9}$
$O_{f_{84}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{2,6,7} \cap \Delta_{3,8,9}$	$m_{n\geq 9}$
$O_{f_{85}^4}$	$\Delta_{1,2,9} \cap \Delta_{1,3,4} \cap \Delta_{1,5,6} \cap \Delta_{2,7,8}$	$m_{n\geq 9}$
$O_{f_{86}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{1,6,7} \cap \Delta_{1,8,9}$	$m_{n\geq 9}$
$O_{f_{87}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,5,6} \cap \Delta_{2,3,9} \cap \Delta_{3,7,8}$	$m_{n\geq 9}$
$O_{f_{88}^4}$	$\Delta_{1,2,4} \cap \Delta_{1,3,5} \cap \Delta_{2,3,9} \cap \Delta_{6,7,8}$	$m_{n\geq 9}$
$O_{f_{89}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,10} \cap \Delta_{4,5,6} \cap \Delta_{7,8,9}$	$m_{n\geq 10}$
$O_{f_{90}^4}$	$\Delta_{1,2,10} \cap \Delta_{1,3,4} \cap \Delta_{2,5,6} \cap \Delta_{7,8,9}$	$m_{n\geq 10}$
$O_{f_{91}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{1,9,10} \cap \Delta_{6,7,8}$	$m_{n\geq 10}$
$O_{f_{92}^4}$	$\Delta_{1,3,4} \cap \Delta_{1,5,6} \cap \Delta_{2,7,8} \cap \Delta_{2,9,10}$	$m_{n\geq 10}$
$O_{f_{93}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,10,11} \cap \Delta_{4,5,6} \cap \Delta_{7,8,9}$	$m_{n\geq 11}$
$O_{f_{94}^4}$	$\Delta_{1,2,3} \cap \Delta_{4,5,6} \cap \Delta_{7,8,9} \cap \Delta_{10,11,12}$	$m_{n\geq 12}$
$O_{f_{95}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,\overline{3}} \cap \Delta_{1,2,4} \cap \Delta_{1,3,4}$	m_4
$O_{f_{96}^4}$	$\Delta_{1,2,3} \cap \Delta_{1,2,\overline{3}} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5}$	m_5

3.4 Proof of the Theorem 3.3.1

Proof: *For $n \geq 12$, the faces of codimension 3 of m_n are partitioned into 15 orbits generated by* $\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,4}$, $\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,\overline{3},4}$, $\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,2,5}$, $\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{1,3,5}$, $\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{3,4,5}$, $\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{\overline{3},4,5}$,

$\triangle_{1,2,3} \cap \triangle_{1,2,4} \cap \triangle_{1,5,6}$, $\triangle_{1,2,3} \cap \triangle_{1,2,4} \cap \triangle_{3,5,6}$, $\triangle_{1,2,3} \cap \triangle_{1,4,5} \cap \triangle_{2,4,6}$, $\triangle_{1,2,3} \cap \triangle_{1,4,5} \cap \triangle_{\overline{2},4,6}$, $\triangle_{1,2,3} \cap \triangle_{1,2,4} \cap \triangle_{5,6,7}$, $\triangle_{1,2,3} \cap \triangle_{1,4,5} \cap \triangle_{1,6,7}$, $\triangle_{1,2,3} \cap \triangle_{1,4,5} \cap \triangle_{2,6,7}$, $\triangle_{1,2,3} \cap \triangle_{1,4,5} \cap \triangle_{6,7,8}$, $\triangle_{1,2,3} \cap \triangle_{4,5,6} \cap \triangle_{7,8,9}$. *Any face of codimension* 4 *of* m_n *can therefore be written as the intersection of a facet* $\triangle$ *of* m_n *with one of these* 15 *faces* $\triangle' \cap \triangle'' \cap \triangle'''$ *of codimension* 3. *If the support* $\sigma(\triangle) \not\subset \{1, ..., 12\}$, *by elementary permutations preserving* $\triangle'$, $\triangle''$, *and* $\triangle'''$ *we can generate* $\widetilde{\triangle} \in O_{\triangle}$ *with* $O_{\triangle' \cap \triangle'' \cap \triangle''' \cap \widetilde{\triangle}} = O_{\triangle' \cap \triangle'' \cap \triangle''' \cap \triangle}$ *and* $\sigma(\widetilde{\triangle}) \subset \{1, ..., 12\}$. *In other words, to generate orbitwise all the subfaces of the canonical faces of codimension* 3 *it is enough to consider the case* $n = 12$. *By applying orbitwise face enumeration algorithm, we can obtain* 94 *orbits of faces of codimension* 4. *Therefore we have to first determine the set* $\mathcal{F}_n(f_i)$ *of facets of* m_n *containing* f_i. *Clearly, if an inequality* (i) *defining a facet of* m_n *is forced to be satisfied with equality by the inequalities defining* $\triangle'$, $\triangle''$, $\triangle'''$, *and* $\widetilde{\triangle}$ *being satisfied with equality, then the same inequality* (i) *- now seen as defining a facets of* m_{n+1} *- will also be forced to be satisfied with equality. In other words, the set* $\mathcal{F}_n(f_i)$ *can only increase with* n *and* $\dim(f_i)$ *can only decrease with* n. *Therefore, only the* 94 *faces of codimension* 4 *for* m_{12} *given in Table are candidates for being faces of codimension* 4 *for* $m_{n\geq 12}$. *A case by case study of the* 94 *faces* f_i, *gives* $\mathcal{F}_n(f_i)$ *and proves that indeed these* 94 *faces generate* 94 *orbits of faces of codimension* 4 *for* $n \geq 12$. *The idea is simply to notice that the pattern of* $\mathcal{F}_n(f_i)$ *is essentially given by the value of* $\mathcal{F}_{15}(f_i)$. *Since all the cases are similar, we only present the computation of* $\mathcal{F}_n(f_{94})$ *where* $f_{94} = \triangle_{1,2,3} \cap \triangle_{4,5,6} \cap \triangle_{7,8,9} \cap \triangle_{10,11,12}$. *Using the orbitwise face enumeration algorithm with t=4, one can easily check that* $\mathcal{F}_{15}(f_{94}) = \{\triangle_{1,2,3}, \triangle_{4,5,6}, \triangle_{7,8,9}, \triangle_{10,11,12}\}$.*Let* $n \geq 15$ *and* $\triangle$ *be a facet of* m_n *with* $\sigma(\triangle) \not\subset \{1, ..., 15\}$. *By elementary permutations preserving* $\mathcal{F}_{15}(f_{94})$ *we* *can generate* $\widetilde{\triangle} \in O_{\triangle}$ *with* $\sigma(\widetilde{\triangle}) \subset \{1, ..., 15\}$. *Let now consider* $\widetilde{\triangle}$ *as a facet of* m_{15}. *Since* $\widetilde{\triangle} \notin \mathcal{F}_{15}(f_{94})$ *at least one vertex* v *of* m_{15} *satisfies* $v \in f_{94}$ *and* $v \notin \widetilde{\triangle}$. *Then, the (*$n$-15*)-times 0-extension* v_{ext} *of* v *is a vertex of* m_n *satisfying* $v_{ext} \in f_{94}$ *but* $v_{ext} \notin \widetilde{\triangle}$ *where* $\widetilde{\triangle}$ *is now considered as a facet of* m_n. *Thus,* $\widetilde{\triangle} \notin \mathcal{F}_n(f_{94})$ *and, by the same elementary permutations,* $\triangle \notin \mathcal{F}_n(f_{94})$*;that is,* $\mathcal{F}_n(f_{94}) = \{\triangle_{1,2,3}, \triangle_{4,5,6}, \triangle_{7,8,9}, \triangle_{10,11,12}\}$ *and* $\operatorname{codim}(f_{94}) = 4$ *for any* $n \geq 12$.

Chapter 4

Implementation and Design

Generating the canonical representatives is one of major computational challenges for the enumeration of the upper layers of the orbitwise face lattice. In our work, we show that by further exploring the combinatorial structures, we can generate most of the canonical representatives efficiently. In this chapter, we present the design of preprocessing heuristics

4.1 Design

In this section, we sketch the orbitwise face enumeration algorithm and present the framework of our preprocessing heuristics. We use the following terminology: matrix representation, row-sum-set, column-sum-set, segment-i and trace of a (-1). A face f^{d-t} can be defined as intersection of some facets of a polytope P; that is, we can define a face f^{d-t} as $f^{d-t} = f_1^{d-1} \cap \ldots \cap f^{d-1}_{||\mathcal{F}(f^{d-t})||}$, where f_i^{d-1} is defined by $a_i^T x = b_i, x \in P$. A face f_r^{d-t} of m_n can be represented by a set of vectors with n entries. For example, the vector representation of the face $f^{d-2} = \triangle_{1,2,3} \cap \triangle_{1,2,4}$, where $n = 4$, is

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

This set of vectors is called as the matrix representation of the face f^{d-t}. The row-sum-set of f^{d-t} is defined as the set $R := \{r_1, r_2, ...\}$, where r_i is the sum of i-th row of the matrix representation. Similarly, the column-sum-set of f^{d-t} is the set $C := \{c_1, c_2, ...\}$, where c_i is the sum of i-th column of the matrix representation. The segment-i refers to the set of columns, of which the sum of all entries of each column equals to i. The trace of a (-1) refers to the collection of other possible entries to become (-1) entries by switchings or multiplying by (-1) row-wise.

Given the list L^{d-t} the orbitwise face enumeration algorithm calls three main subroutines to obtain the canonical representative list $\mathcal{L}^{d-t}$; (i) generation of the set $\{\tilde{f}^{d-t+1} \cap f^{d-1}\}$, (ii) computation of the facet set $\mathcal{F}(\tilde{f}^{d-t+1} \cap f^{d-1})$, and (iii) identifying the elements belonging to $\mathcal{L}^{d-t}$. Note that to obtain $\mathcal{L}^{d-t}$, it is not necessary to compute the facet set for each output element from first subroutine since some of the output elements can be identified as belonging to the same orbits even if without the complete facet set. A face of codimension t can be defined by t equalities. Another element of the algorithm is to perform efficiently orbitwise equivalency checks and this is the core of our designed preprocessing heuristics.

1. Orbitwise invariants check: Set of invariants to differentiate faces into partitions, such as number of cuts, anti-cuts, and trivial-extension-cuts.

2. Orbitwise equivalency check: Exploiting the matrix representation to identify if there exists an isometry between faces.

Computing the set $\mathcal{F}(\tilde{f}_s^{d-t+1} \cap f_r^{d-1})$ requires to call the linear programming subroutine $O(m)$ times, where m is the number of facets. Given pairs of dimension-$(d-t)$ faces from the list L^{d-t}, we can look either for invariants proving that the faces are not orbitwise equivalent or permutation and switching to prove they are orbitwise equivalent. While the invariants are used to partition faces into subsets and, therefore, provide a lower bound for $|\mathcal{L}^{d-t}|$, checking the orbitwise equivalency provides a upper bound for $|\mathcal{L}^{d-t}|$.

One important issue when developing the heuristics is the trade-off between the computational cost and the effectiveness to obtain a certificate of equivalency or non-equivalency. Figure 4.1 outlines the order in which the heuristics are performed.

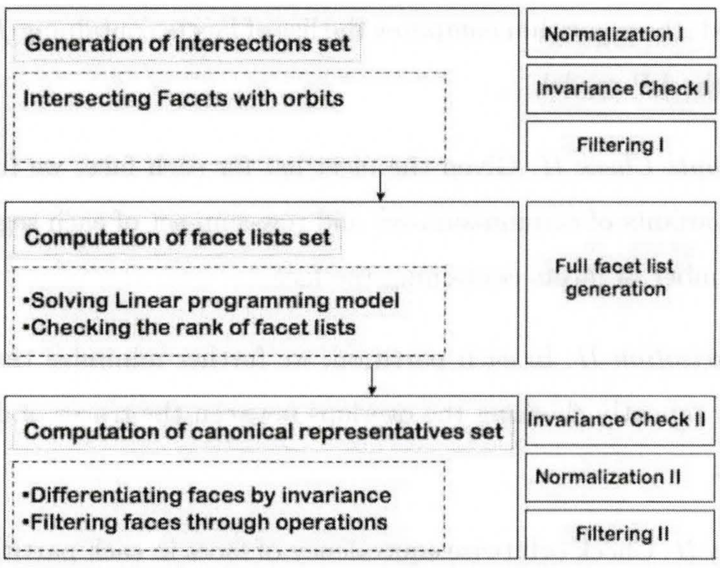

Figure 4.1: Sequential heuristics

- *Regularization I*: Apply permutation and switching operations to all the elements in L^{d-t} to put 0 entries after -1 or 1 entries, then minimize the overall number of -1.

- *Invariants Check I*: Check the orbitwise invariants: the number of cuts, anti-cuts, and trivial-extension-cuts. Note that at this stage we do not need to compute the facet set $\mathcal{F}(\tilde{f}_s^{d-t+1} \cap f_r^{d-1})$ for each face.

- *Tuning I*: Check orbitwise equivalency of faces in each partition by temporarily treating -1 as 1.

- *Facet set $\mathcal{F}(\tilde{f}^{d-t+1} \cap f^{d-1})$ Generation*: As the lower bound for $|\mathcal{L}^{d-t}|$ is reduced, the algorithm computes the list of facets containing $\tilde{f}^{d-t+1} \cap f^{d-1}$ using the LP model.

- *Invariants Check II*: Given the facet list for each face, we further check the invariants of column-sum-set and row-sum-set of each segment-i, and the number of facets containing the face.

- *Regularization II*: In each partition, we further minimize the number of (-1) by repeatly checking the overlaps between the traces of different (-1) entries.

- *Tuning II*: Check orbitwise equivalency of faces in each partition by using the information of the traces of (-1)s.

- *Exhaustive-Invariants-Check*: Full generation of the action of the symmetry group $\mathrm{Is}(P)$.

For *codimension* ≤ 4 we successfully generate all the orbits without using Exhaustive-Invariants-Check approach. The order of execution is set according to the empirical performance of the heuristics. For example, the heuristic Regularization I empirically identifies a much larger number of candidates belonging to the same orbits than other heuristics, and is therefore performed first.

4.2 Implementation for the metric polytope on 12 nodes

We provide the first complete orbitwise description of the faces of codimension 4 for any n of the metric polytope. In this section, we give a detail discussion of the performance of different heuristics applied to compute the canonical representatives of codimension 4 of m_{12}. We also present the few faces which challenge the heuristics and require further investigation.

Figure 1 *Enumeration of faces of codimension* 4 *for* m_{12}

steps	lower bound for $\|\mathcal{L}^{d-4}\|$	upper bound for $\|\mathcal{L}^{d-4}\|$
Generating L^{d-4}	1	13155
Regularization I	1	1186
Invariants check I	37	1186
Tuning I	37	300
Invariants check II	92	300
Regularization II	92	167
Tuning II	92	98
Special cases filtering	92	94
Exhuastive checking	94	94

Throughout the steps of heuristics, the algorithm decreases the gap between the upper and lower bounds for $|\mathcal{L}^{d-4}|$, and as it reaches zero, the enumeration of all the orbits is complete. Our designed heuristics successfully bring the gap from 13154 to 6 till the step Tuning II. In the step Special cases filtering, the undetermined 6 faces in L^{d-4} are examined by first applying switching operation or multiplying by (-1) row-wise even if the total number of (-1) increase to enhance the possibility to identify equivalency. The step successfully identifies 4 of those 6 faces as equivalent to some of the 92 faces forming the lower bound for $|\mathcal{L}^{d-4}|$. Finally, the exhaustive approach is applied to identify the remaining two elements forming two new orbits.

Figure 2 *Four elements solved by special cases filtering and the orbits they belong to:*

faces require further investigation for Is(P)-equivalency
$O_{\tilde{f}^4_4}$, $\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{\overline{2},4,6} \cap \Delta_{\overline{3},5,6}$
$O_{\tilde{f}^4_{13}}$, $[\Delta_{1,2,3} \cap \Delta_{1,\overline{2},4} \cap \Delta_{1,3,4} \cap \Delta_{2,3,5}]$, $[\Delta_{1,2,3} \cap \Delta_{1,\overline{2},4} \cap \Delta_{1,3,4} \cap \Delta_{2,5,6}\ ]$
$O_{\tilde{f}^4_{14}}$, $\Delta_{1,2,3} \cap \Delta_{1,2,\overline{4}} \cap \Delta_{1,3,4} \cap \Delta_{2,5,6}$

As the earlier steps heuristics try to minimize the number of (-1) by multiplying rows or columns by (-1) so that the number of (-1) is reduced. This greedy approach fails for the above 4 faces. To tackle these 4 faces, we multiply rows or columns by (-1) even if the number of (-1) increases.

Example 1 *The canonical representative $O_{\tilde{f}^4_{13}}$ is defined as:*

$$\mathcal{F}(\tilde{f}^4_{13}) := \begin{bmatrix} 1 & 0 & -1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 & 0 & 0 \end{bmatrix} \quad \textit{and}$$

$$\mathcal{F}(\Delta_{1,2,3} \cap \Delta_{1,\bar{2},4} \cap \Delta_{1,3,4} \cap \Delta_{2,3,5}) := \begin{bmatrix} -1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & -1 & 0 & 0 \end{bmatrix}$$

and it can be checked that f_{13}^4 and $\Delta_{1,2,3} \cap \Delta_{1,\bar{2},4} \cap \Delta_{1,3,4} \cap \Delta_{2,3,5}$ are in the same orbit by permuting the 3rd and 5th rows of $\mathcal{F}(\Delta_{1,2,3} \cap \Delta_{1,\bar{2},4} \cap \Delta_{1,3,4} \cap \Delta_{2,3,5})$ and then multiplying 1st row and 5th column by (-1).

Figure 3 *Two elements solved by exhuastive checking,*

Representative	faces requiring further investigation
$\Delta_{1,2,3} \cap \Delta_{1,4,5} \cap \Delta_{\bar{2},4,6} \cap \Delta_{3,\bar{5},6}$	$\Delta_{1,2,3} \cap \Delta_{1,2,4} \cap \Delta_{3,5,6} \cap \Delta_{\bar{4},5,6}$
$\Delta_{1,2,3} \cap \Delta_{1,4,6} \cap \Delta_{2,5,7} \cap \Delta_{\bar{3},4,5}$	$\Delta_{1,2,6} \cap \Delta_{1,3,4} \cap \Delta_{2,5,7} \cap \Delta_{3,4,\bar{5}}$

We identify the last 2 faces as forming 2 new orbits by checking the row-sum-set for each segment-i of the two elements and the respective canonical representatives of the partitions they belong to, treating (-1) as 1. We found that there is no feasible row-permutation to make the row-sum-set for each segment-i of $\mathcal{F}(f_i^{d-t})$ and $\mathcal{F}(f_j^{d-t})$ be equivalent.

4.3 Orbitwise invariants check

In this section detailed description of orbitwise invariants check and respective heuristics are presented. Invariants refer to the quantity which remains unchanged within certain orbit. The algorithm often checks whether two faces f_i and f_j are Is(P)-equivalent; that is, whether they belong to the same orbit under the action of $\text{Is}(P)$ or not. If any invariant is different for f_i and f_j, the faces must be in different orbits. The codimension of a face and its cardinality are easily checkable invariants. Some other invariants can easily be added, for example by looking at the action of $\text{Is}(P)$ on pairs, triples, or other $k-$tuples of indices (generators), respectively on lower dimensional faces. The number of elements from each such orbit included in a face is a $\text{Is}(P)$-invariant. In the implementation, the number of cuts, anti-cuts, and trivial-extension-cuts invariants are checked in Invariants check I. The number of incident facets, minimum support and the row-sum-set of each segment-i invariants are checked in Invariants check II. One major reason to implement the orbitwise invariants check separately is that Invariants check I does not require to have the complete facet set $\mathcal{F}(f_r^{d-t})$ while it is required for Invariants check II. The number of calls to linear programming subroutine when computing the complete facet set $\mathcal{F}(f_r^{d-t})$ is reduced. Figure 1 in section 4.2 shows that the number of calls to linear programming subroutine is reduced to 300 from 1186. We give a bit more emphasis on the invariant, row-sum-set of each segment-i of a face as this invariant appears to be efficient. By utilizing the invariant, each time we compare only part of the matrix representations(segment-i) between faces. In some cases, the $\text{Is}(P)$-in-equivalency between faces is obtained without considering the full matrix representation. The following example demonstrates how to find for invariants by looking at the action of $\text{Is}(P)$ on a face.

Example 2 *Given complete facet set,*

$$\mathcal{F}(f_i^{d-4}) := \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

We examine the action of row permutation on $\mathcal{F}(f_i^{d-4})$. The action changes the fill-ins of each column. However, it does not change the sum of all the entries of each column. It follows that the column-sum-set of a face is an invariant. The column-sum-set of $\mathcal{F}(f_i^{d-4})$ is $\begin{pmatrix} 3 & 3 & 2 & 2 & 1 & 1 \end{pmatrix}$. Similarly, the column permutation does not change the sum of all the entries of each row. A more useful observation is that the column permutation does not change the row-sum-set of each segment-i. Therefore the row-sum-set of each segment-i is also an invariant. For example, the row-sum-set of segment-3 for $\mathcal{F}(f_i^{d-4})$ is $\begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}$. For a face to be orbitwise-equivalent to f_i^{d-4}, it is necessary to have the same row-sum-set of its segment-3.

Remark 4.3.1 *For m_{12} and codimension 4, the proposed segment-based row-sum-set check distinguished 92 orbits among 94 final orbits. For $n \leq 8$, and codimension ≤ 4, the method distinguishes all the orbits.*

4.3.1 Computation of the facet set of a face

Assuming $\tilde{f_s}^{d-t+1} = \{f_1^{d-1}, ..., f_{t-1}^{d-1}\}$ and $f_r^{d-1} = f_t^{d-1}$ by re-ordering the given facets and defining $f_i^{d-1} = \{a_i^T x = b_i, x \in m_n\}$, we implement the following modified linear programing model to compute the facet set $\mathcal{F}(\tilde{f_s}^{d-t+1} \cap f_r^{d-1})$

$$\begin{array}{lll} \max & \sum_{j=t+1}^{m}(a_j^T x - b_j) & \\ \text{s.t.} & a_i^T x - b_i = 0 & i = 1, ..., t \\ & a_i^T x - b_i \geq 0 & i = t+1, ..., m \\ & x \in I\!R^{\binom{n}{2}} & \end{array}$$

In the model, the objective function is the sum of all the slack variables except the ones corresponding to the facet set incident to the face $(\tilde{f}_s^{\,d-t+1} \cap f_r^{d-1})$. Applying this model lowers the number of calls to the linear programing solver, however, it requires perturbation testing over 0-valued solution entries as a facet detaching from the face $(\tilde{f}_s^{\,d-t+1} \cap f_r^{d-1})$ could have 0 as solution of the LP.

4.4 Orbitwise equivalency check

We present the heuristics developed to identify the orbitwise equivalency by utilizing the matrix representation: Regularization class and Tuning class. For regularization class, the main idea is to systematically arrange 0, 1,-1 entries and reduce -1 entries of all candidates of orbitwise faces. A large number of faces can be identified this way as equivalent. For Tuning class, the heuristics tend to "tune" one face to another; that is, to arrange 0,1,-1 entries of one face in order to approximate the matrix representation of the other. One way to tune one face to another is to reorder the rows so that the row-sum-set of each segment-i is equivalent. Even though the equivalency of row-sum-set of each segment-i of two faces is not sufficient to prove the orbitwise equivalency of two faces, the tuning helps to check the orbitwise equivalency. Similar ideas are applied in the heuristics introduced in the following sections.

4.4.1 Regularization I

In Regularization I, the heuristic permutes the columns of matrix representation so that the columns are in the increasing order with respect to the sum of all entries in a column. Next, the heuristic minimizes the number of (-1) of each column. After applying this regularization procedure over all candidates of orbitwise faces, the heuristic tries to identify the equivalency of matrix representations among faces. Empirically, the heuristic succeeds in identifying a large number of orbitwise-equivalent faces.

4.4.2 Tuning I

Tuning-class heuristics try to tune one face to another to check orbitwise equivalency. To check the orbitwise equivalency of two faces f_i and f_j, the Tuning I permutes the rows of the matrix representation of f_i so that the row-sum-set of each segment-i of f_i can approximate the one of f_j. For example, if the row-sum-set of segment-i of f_i is (0,1,2,3), in order to satisfy the equivalency of the row-sum-set of segment-i between f_i and f_j, there is only one permutation to consider. It follows that the heuristic permutes rows by first considering the segment for which the row-sum-set has the most distinguished values.

Example 3 *We illustrate the idea of permuting rows by first considering the segment for which the row-sum-set has the most distinguished values.*

Given a face f_i^{d-4}: $\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$, *the row-sum-set of segment-4 is* $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$. *The row-sum-set of segment-3 is* $\begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}$ *and the row-sum-set of segment-1 is* $\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$. *We can either look at segment-3 or segment-1 as the segment for which the row-sum-set has the most distinguished values. There are only 3 possible* *row permutations:* $\{1,4\},\{2,3\},\{(1,4),(2,3)\}$

Another illustration using the face $f_k^{d-4} := \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$

the row-sum-set of segment-2 is $\begin{pmatrix} 3 \\ 2 \\ 1 \\ 0 \end{pmatrix}$. *There is only one possible permutation by looking at segment-2.*

When the equivalency of the row-sum-set of each segment could be attained by row permutations, the next step is to check the matching between the set of columns within two matrix representations. It is enough to determine the orbitwise equivalency of two faces by exhibiting a bijection between two sets of columns. The heuristic separately checks the bijection between the sets of columns for each segment. For codimension ≤ 4 the bijection between the sets of columns of the segment usually holds if the row-sum-set of certain segment were equivalent between two faces.

Proposition 4.4.1 *For a face of codimension* 4*, given the row-sum-set of segment-i* $\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}$, *where* $i = 1,3,4$*, the set of columns of respective segment-i is unique. For segment-2, given the number of the column* $v_1 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$ *and* $v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$, *the set of columns of segment-2 is also unique.*

Proof: For a face of codimension 4, the uniqueness of the set of columns for segment-1, 3, 4 could be proved similarly as for codimension 3. For segment-2, the possible columns are $\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$.

Given the row-sum-set of segment-2 $\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}$ *and the number of column v_1, v_2, it can be shown that there exists unique solution for the number of the remaining 4 vectors by checking the rank.*

4.4.3 Regularization II

When regularizing faces, a common task is to reduce (-1) entries of matrix representation. In Regularization I, the heuristic reduce (-1) entries column-wise. In Regularization II, the heuristic tries to reduce (-1) entries both column-wise and row-wise. In practice, we apply the switching operation and multiplying (-1) row-wise to change (-1) entries. The following example demonstrates how the (-1) entries can be reduced.

Example 4 *Given two faces f_i^{d-4} and f_j^{d-4} with different settings of fill-ins:*

$$f_i^{d-4}: \begin{bmatrix} 1 & 0 & 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$f_j^{d-4}: \begin{bmatrix} 1 & 0 & 1 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

We show how to reduce (-1) entries of f_i^{d-4}

$$f_i^{d-4}: \begin{bmatrix} 1 & 0 & 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

multiply 4th row by (-1)$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 \end{bmatrix}$

multiply 5th column by (-1)$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 \end{bmatrix}$

We can also reduce (-1) entries of f_j^{d-4} *as follows:*

$$f_j^{d-4}: \begin{bmatrix} 1 & 0 & 1 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

multiply 5th column, 4th row, 6th column by (-1) in turn$\Rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$

multiply 3rd row by (-1)$\Rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$

$$\textit{multiply 4th column by (-1)}\Rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The two faces can be identified as equivalent by proper permutations.

As in the above example, tracing all the (-1) entries after applying each operation helps to identify the proper operations to further reduce (-1) entries, see Figure 4.2.

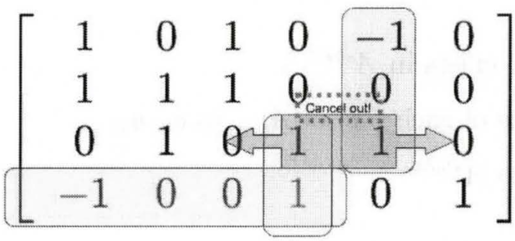

Figure 4.2: Regularization II

The heuristic repeatedly checks the trace of (-1) entries until no more reduction can be found. The outline is as follows:

Algorithm: Regularization II

Input: A_0: matrix representation of a face f_i

Output: A_{after}: matrix representation of f_i after (-1) reduction

Put A_0 in a stack: A^{stack}

Save the trace of each (-1) of A_0 to the matrices a_0, a_1 ,...

While (The top element A_i in A^{stack} is not yet checked for (-1) reduction)

  Compare a_0, a_1, ... to see if the number of (-1) could be reduced

  if (the number of (-1) could be reduced)

    Generate A_{i+1} from A_i after (-1) reduction

  end

  if(A_{i+1} not yet exists in A^{stack})

    Save the trace of each (-1) of A_{i+1} to $a_0, a_1, ...$

    Put A_{i+1} unto A^{stack}

  end

end

Output the top element A_{after} in A_{stack}

4.4.4 Tuning II

Tuning II heuristic is designed to check orbitwise equivalency of two faces having equivalent column-sum-set and row-sum-set of each segment when treating (-1) as 1, and two faces have different (-1) entries. The heuristic tends to tune one face to another. It applies permutation/switching operations to change (-1) entries of one face to approximate the (-1) entries of the other as illustrated by the following example.

Example 5 *Given the two faces f_i^{d-4} and f_j^{d-4}:*

$$f_i^{d-4}: \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$f_j^{d-4}: \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The algorithm applies the following actions on f_i^{d-4}:

multiply third column by (-1), and multiply first row by (-1)$\Rightarrow$ $\begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

Next, permute 1st, 4th row, and permute 2nd, 3rd row$\Rightarrow$ $\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$

Finally, permute 1st, 2nd column, then 3rd, 4th column, and 5th, 6th column $\Rightarrow f_j^{d-4}$

The heuristic first change (-1) entry to the entry of specific column. The column must have the same sum of all its entries as that of the column of another face containing the (-1) entry. The heuristic checks if the trace of (-1) under switching or multiplying (-1) row-wise overlap the column, see Figure 4.3 for an illustration.

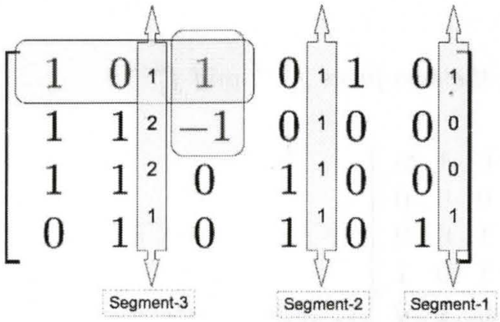

Figure 4.3: Tuning II

When possible, the heuristic checks if there exists row permutation to change (-1) entry to the same (-1) entry of another face. The outline is as follows.

Algorithm: Tuning II

Input: f_i, f_j: matrix representations of two faces

Output: indicator showing if f_i, f_j are orbitwise-equivalent

$R_{f_j} \leftarrow$ the rows to permute without changing f_j

$C_{f_j} \leftarrow$ the columns overlapping the trace of any (-1) in f_j

for $a_0 :=$ an entry $\in$ {traces of all (-1) in f_i }

    if $a_0 \in R_{f_j} \times C_{f_j}$

        $f_i \leftarrow f_i$ after moving (-1) to a_0

        $f_j \leftarrow f_j$ after moving (-1) to a_0

```
      if( f_i == f_j )
         return "f_i, f_j are orbitwise-equivalent"
      end
   end
end
return "unknown";
```

4.4.5 Dealing with faces belonging to many facets

The large size of $\mathcal{F}(f_r^{d-t})$ may cause some difficulty to our method as it increases the input size for many of our heuristics. We may hope that our proposed invariants could be as efficient as in codimension 4 case to partition the input into subsets, and design the following heuristic. Instead of matching two full facet sets, $\mathcal{F}(f_i^{d-t})$ and $\mathcal{F}(f_j^{d-t})$, the heuristic selects k facets among $\mathcal{F}(f_i^{d-t})$ or $\mathcal{F}(f_j^{d-t})$ to perform the matching. If the matching is unsuccessful, the heuristic remove one facet among the selected k facets, and add one facet among the remaining facets and try again to match.

Chapter 5

Conclusion

This thesis deals with the face lattice enumeration problem for convex polytope in general dimension, focusing on polytopes arising from combinatorial optimization problem. In particular, we study the metric polytope associated to the well-known maxcut and multicommodity flow problems, as well as to finite metric space. Exploiting the high degree of symmetry, we provide the first complete orbitwise description of the faces of codimension 4 of the metric polytope for any dimension. The full face lattice is computed for small instances. While the following layers of the upper face lattice probably require advanced computations on a parallel cluster, the orbitwise description of the faces of codimension 4 was achieve through a combination of partitions using orbitwise invariants and heuristics to identify equivalency permutations and switchings.

Bibliography

[1] D. Avis, D. Bremner and R. Seidel: How Good are Convex Hull Algorithms? Computational Geometry: Theory and Applications **7** (1997) 265-301.

[2] D. Avis and F. Fukuda: A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. Discrete and Computational Geometry **8** (1992) 295-313.

[3] D. Avis, K. Fukuda and S. Picozzi: On the canonical representations of convex polyhedra. In: A. M. Cohen, X. S. Gao, N. Tokuyama (eds.): Mathematical Software, World Scientific (2002) 351-360.

[4] M. L. Balinski: On the graph structure of convex polyhedra in n-space. Pacific Journal Mathematics **11** (1961) 431-434.

[5] M. Bayer and C. Lee: Combinatorial aspects of convex polytopes. In P. Gruber and J. Wills (eds.) Handbook on Convex Geometry, North Holland (1994) 485-534.

[6] R. Blind and P. Mani-Levitska: On puzzles and politope isomorphisms. Aequationes Mathematics **34** (1987) 287-297.

[7] D. BREMNER, M. DUTOUR SIKIRIĆ AND A. SCHÜRMANN : Polyhedral representation conversion up to symmetries. submitted (2006).

[8] A. BRØNDSTED: An Introduction to Convex Polytopes. Graduate Texts in Mathematics 90 Springer-Verlag (1983).

[9] A. BROUWER, A. COHEN AND A. NEUMAIER: Distance-Regular Graphs. Springer-Verlag (1989).

[10] D. CHAND AND S. KAPUR: An algorithm for convex polytopes. Journal of Association for Computing Machinery **17** (1970) 78-86.

[11] A. DEZA AND M. DEZA: The ridge graph of the metric polytope and some relatives. In: T. Bisztriczky, P. McMullen, R. Schneider, A. I. Weiss (eds.): Polytopes: Abstract, Convex and Computational (1994) 359-372.

[12] A. DEZA AND M. DEZA: The combinatorial structure of small cut and metric polytopes. Combinatorics and Graph Theory 95 World Scientific, Singapore (1995) 70-88.

[13] A. DEZA, K. FUKUDA, T. MIZUTANI AND C. VO: On the face-lattice of the metric polytope. Lecture Notes in Computer Science 2866 Springer-Verlag, Berlin (2003) 118-128.

[14] M. DEZA, V. GRISHUKHIN AND M. LAURENT: The symmetries of the cut polytope and of some relatives. In: P. Gritzmann and B. Sturmfels (eds.) Applied Geometry and Discrete Mathematics, the Victor Klee Festschrift DIMACS Series in Discrete Mathematics and Theoretical Computer Science **4** (1991) 205-220.

[15] A. Deza and G. Indik: A counterexample to the dominating set conjecture. Optimization Letters 1-2 (2007) 163-169.

[16] M. Deza and M. Laurent: Geometry of cuts and metrics. Algorithms and Combinatorics **15** Springer (1997) 403-406.

[17] K. Fukuda and A. Prodon: Double description method revisited. Lecture Notes in Computer Science **1120** Springer-Verlag (1996) 91-111.

[18] K. Fukuda and V. Rosta: Combinatorial face enumeration in convex polytopes. Computational Geometry: Theory and Applications **4** (1994) 191-198.

[19] B. Grünbaum: Convex Polytopes. Wiley (1967).

[20] G. Kalai: A simple way to tell a simple polytope from its graph. Journal of Combinatorial Theory Series A **49** (1988) 381-383.

[21] M. Laurent and S. Poljak: The metric polytope. In: E. Balas, G. Cornuejols and R. Kannan (eds.): Integer Programming and Combinatorial Optimization (1992) 247-286.

[22] T. S. Motzkin, H. Raiffa, G. L. Thompson and R. M. Thrall: The double description method. In H. W. Kuhn and A. W. Tucker (eds.) Annals of Mathematics Studies **28** Princeton University Press (1953) 51-73.

[23] P. McMullen and G. C. Shephard: Convex Polytopes and the Upper Bound Conjecture. London Mathematical Society Lecture Note Series 3 Cambridge University Press (1971).

[24] R. SEIDEL: A convex hull algorithm optimal for point sets in even dimensions. University of British Columbia, Vancouver (1981).

[25] A. SCHRIJVER: Theory of Linear and Integer Programming. John Wiley and Sons, New York (1986).

[26] E. STEINITZ AND H. RADEMACHER: Vorlesungen über die Theorie der Polyeder. Berlin, Springer-Verlag (1934) reprint, New York-Heidelberg-Berlin, Springer-Verlag (1976).

[27] G. M. ZIEGLER: Lectures on Polytopes. Graduate Texts in Mathematics **152** Springer-Verlag (1995).

7549 4