Enumeration of 2-Level Polytopes

Adam Bohn[1], Yuri Faenza[2], Samuel Fiorini[1], Vissarion Fisikopoulos[1], Marco Macchia[1], and Kanstantsin Pashkovich[3]

[1] Université libre de Bruxelles, Brussels, Belgium
{adam.bohn,sfiorini,vfisikop,mmacchia}@ulb.ac.be
[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
yuri.faenza@epfl.ch
[3] C & O Department, University of Waterloo, Waterloo, Canada
kanstantsin.pashkovich@gmail.com

Abstract. We propose the first algorithm for enumerating all combinatorial types of 2-level polytopes of a given dimension d, and provide complete experimental results for $d \leqslant 6$. Our approach is based on the notion of a simplicial core, that allows us to reduce the problem to the enumeration of the closed sets of a discrete closure operator, along with some convex hull computations and isomorphism tests.

Keywords: Polyhedral computation, Optimization, Formal concept analysis

1 Introduction

A (convex) polytope $P \subseteq \mathbb{R}^d$ is said to be 2-*level* if for every facet-defining hyperplane H, there exists another hyperplane H' parallel to H which contains all the vertices of P that are not contained in H.

There are a number of alternative ways to define 2-level polytopes. For example, a polytope P is said to be *compressed* if every pulling triangulation of P is unimodular with respect to the lattice generated by its vertices [18,11,5]. In [20] this property is shown to be equivalent to 2-levelness. Given a finite set $V \subseteq \mathbb{R}^d$ and a positive integer k, the k-th *theta body* of V is a tractable convex relaxation of the convex hull of V. The *theta rank* of V is defined as the smallest k such that this relaxation is exact. These notions were introduced in [8] in a more general context in which V can be the set of real solutions of any finite system of real polynomials. The authors of [8] show that a finite set has theta rank 1 if and only if it is the vertex set of a 2-level polytope.

Families of 2-level polytopes appear in a number of different combinatorial contexts: Birkhoff polytopes, Hanner polytopes [12], stable set polytopes of perfect graphs [4], Hansen polytopes [13], order polytopes [19] and spanning tree polytopes of series-parallel graphs [10] all have the 2-level property. Because they appear in such a wide variety of contexts, 2-level polytopes are interesting objects. However, our understanding of them remains relatively poor. In this paper we study the problem of enumerating all combinatorial types of 2-level polytopes of a fixed dimension.

N. Bansal and I. Finocchi (Eds.): ESA 2015, LNCS 9294, pp. 191–202, 2015.
DOI: 10.1007/978-3-662-48350-3_17

Since every 2-level polytope is affinely equivalent to a 0/1-polytope, one might think to compute all those of a given dimension simply by enumerating all 0/1-polytopes of that dimension and discarding those which are not 2-level polytopes. However, the complete enumeration of d-dimensional 0/1-polytopes has been implemented only for $d \leqslant 5$ [1]. The same author has enumerated all those 6-dimensional 0/1-polytopes having up to 12 vertices, but the complete enumeration even for this low dimension is not expected to be feasible: the output of the combinatorial types alone is so huge that it is not currently possible to store it or search it efficiently [22]. Thus for all but the lowest dimensions, there is no hope of working with a pre-existing list of 0/1-polytopes, and it is desirable to find an efficient algorithm which computes 2-level polytopes from scratch.

1.1 Contribution and Outline

We present the first algorithm to enumerate all combinatorial types of 2-level polytopes of a given dimension d. The algorithm uses new structural results on 2-level polytopes which we develop here.

Our starting point is a pair of full-dimensional embeddings of a given 2-level d-polytope that are related to each other via some $d \times d$ unimodular, lower-triangular 0/1-matrix. This is explained in Section 3. In one embedding, which we refer to as the $\mathcal{H}$-embedding, the facets have 0/1-coefficients. In the other – the $\mathcal{V}$-embedding – the vertices have 0/1-coordinates. The $\mathcal{H}$- and $\mathcal{V}$-embeddings are determined by a structure, which we call a simplicial core (see Section 3.2)

Our algorithm is described in detail in Section 4. It computes a complete list L_d of non-isomorphic 2-level d-polytopes, from a similar list L_{d-1} of 2-level $(d-1)$-polytopes. In these lists, each polytope is stored via its slack matrix (see Section 3.1).

For some polytope $P_0 \in L_{d-1}$, define $L(P_0)$ to be the collection of all 2-level polytopes that have P_0 as a facet. Then the union of these collections $L(P_0)$ over all polytopes $P_0 \in L_{d-1}$ is our desired set L_d, because every facet of a 2-level polytope is 2-level. We proceed as follows: given some $P_0 \in L_{d-1}$, we realize it in the hyperplane $\{x \in \mathbb{R}^d \mid x_1 = 0\} \simeq \mathbb{R}^{d-1}$. We compute a collection $\mathcal{A} \subseteq \{x \in \mathbb{R}^d \mid x_1 = 1\}$ of point sets, such that for each 2-level polytope $P \in L(P_0)$, there exists $A \in \mathcal{A}$ with $P \simeq \mathrm{conv}(P_0 \cup \{e_1\} \cup A)$. For each $A \in \mathcal{A}$, we compute $P = \mathrm{conv}(P_0 \cup \{e_1\} \cup A)$ and, in case it is 2-level and not isomorphic to any of the polytopes already generated by the algorithm, we add P to the list L_d. The efficiency of this approach depends greatly on how the collection $\mathcal{A}$ is chosen. Here, we exploit the pair of embeddings to define a proxy for the notion of 2-level polytopes in terms of closed sets with respect to a certain discrete closure operator, and use this proxy to construct a suitable collection $\mathcal{A}$. This turns out to provide a significative speedup in the computations.

We implemented this algorithm and ran it to obtain L_d for $d \leqslant 6$. The outcome of our experiments is discussed in Section 5. We found that the number of combinatorial types of 2-level d-polytopes is surprisingly small for low dimensions d. Moreover, low-dimensional 2-level polytopes can be used to understand the structure of higher-dimensional 2-level polytopes. For instance, they show which polytopes can appear as low-dimensional faces of higher-dimensional 2-level polytopes.

We conclude the paper by discussing one conjecture inspired by our experiments, and some ideas for future work (see Section 6).

1.2 Previous Work

The problem closest to the one which we study here is that of enumerating 0/1-polytopes, see [1,22]. In our approach, we use techniques from formal concept analysis, in particular we use a previously existing algorithm to enumerate all concepts of a relation, see [6,15]. Some general properties of 2-level polytopes are established e.g., in [20] and [8].

2 Preliminaries

We list here a number of definitions and properties used throughout the paper. For basic notions on polytopes that do not appear here, we refer the reader to [21]. Given a positive integer d, we set $[d] := \{1, \dots, d\}$. A d-polytope is a polytope of dimension d. For $x \in \mathbb{R}^d$ and $E \subseteq [d]$, we let $x(E) := \sum_{i \in E} x_i$.

While in general two polytopes can be combinatorially isomorphic without being affinely isomorphic, for 2-level polytopes these two notions coincide. This is not difficult to see but requires some definitions, so we defer it to Section 3.1 (see Lemma 1). We then say that two 2-level polytopes are *isomorphic* if and only if they are combinatorially isomorphic. A condition stronger than isomorphism is congruency: two polytopes are *congruent* if there is an isometry mapping one to the other.

The *f-vector* of a d-polytope P is the d-dimensional vector whose i-th entry is the number of $(i-1)$-dimensional faces of P. Thus $f_0(P)$ gives the number of vertices of P, and $f_{d-1}(P)$ the number of facets of P. We use $\mathrm{vert}(P)$ to denote the vertex set of polytope P.

3 Embeddings

3.1 Slack Matrices and Slack Embeddings

The *slack matrix* of a polytope $P \subseteq \mathbb{R}^d$ with m facets $F_1, \dots, F_m$ and n vertices $v_1, \dots, v_n$ is the $m \times n$ nonnegative matrix $S = S(P)$ such that S_{ij} is the *slack* of the vertex v_j with respect to the facet F_i, that is, $S_{ij} = g_i(v_j)$ where $g_i : \mathbb{R}^d \to \mathbb{R}$ is any affine form such that $g_i(x) \geqslant 0$ is valid for P and $F_i = \{x \in P \mid g_i(x) = 0\}$. The slack matrix of a polytope is defined up to scaling its rows by positive reals.

The slack matrix provides a canonical way to embed any polytope, which we call the *slack embedding*. This embedding maps every vertex v_j to the corresponding column $S^j \in \mathbb{R}^m_+$ of the slack matrix $S = S(P)$. Every polytope is affinely isomorphic to the convex hull of the columns of its slack matrix.

Due to definition a polytope P is 2-level if and only if $S(P)$ can be scaled to be $0/1$. Given a 2-level polytope, we henceforth always assume that its facet-defining inequalities are scaled so that the slacks are $0/1$. Thus, the slack embedding of a 2-level polytope depends only on the *support* of its slack matrix, which only depends on its combinatorial structure. The next lemma follows from this observation.

Lemma 1. *Two 2-level polytopes are affinely isomorphic if and only if they have the same combinatorial type.*

3.2 Simplicial Cores

A *simplicial core* for a d-polytope P is a $(2d+2)$-tuple $(F_1, \dots, F_{d+1}; v_1, \dots, v_{d+1})$ of facets and vertices of P such that each facet F_i does not contain vertex v_i but contains vertices $v_{i+1}, \dots, v_{d+1}$.

Every d-polytope P admits a simplicial core and this fact can be proved by a simple induction on the dimension, see, e.g., [9, Proposition 3.2]. Actually, simplicial cores for P correspond to $(d+1) \times (d+1)$ submatrices of $S(P)$ that are invertible and lower-triangular, for some ordering of rows and columns.

Notice that, for each i, the affine hull of F_i contains v_j for $j > i$, but does not contain v_i; thus the vertices of a simplicial core are affinely independent. That is, $v_1, \dots, v_{d+1}$ form the vertices of a d-simplex contained in P.

3.3 $\mathcal{H}$- and $\mathcal{V}$-Embeddings

Although canonical, the slack embedding is never full-dimensional, which can be a disadvantage. To remedy this, we use simplicial cores to define two types of embeddings that are full-dimensional. Let P be a 2-level d-polytope with m facets and n vertices, and let $\Gamma := (F_1, \dots, F_{d+1}; v_1, \dots, v_{d+1})$ be a simplicial core for P.

From now on, we assume that the rows and columns of the slack matrix $S(P)$ are ordered compatibly with the simplicial core, so that the i-th row of $S(P)$ corresponds to facet F_i for $1 \leqslant i \leqslant d+1$ and the j-th column of $S(P)$ corresponds to vertex v_j for $1 \leqslant j \leqslant d+1$.

The $\mathcal{H}$*-embedding* with respect to Γ is defined by mapping each v_j to the unit vector e_j of $\mathbb{R}^d$ for $1 \leqslant j \leqslant d$, and v_{d+1} to the origin. In the $\mathcal{H}$-embedding of P, facet F_i for $1 \leqslant i \leqslant m$ is defined by the inequality $\sum_{j \in [d], S_{ij}=1} x_j \geqslant 0$ if $v_{d+1} \in F_i$ and by $\sum_{j \in [d], S_{ij}=0} x_j \leqslant 1$ if $v_{d+1} \notin F_i$.

In the $\mathcal{V}$*-embedding* of P with respect to Γ, vertex v_j is the point of $\mathbb{R}^d$ whose i-th coordinate is S_{ij}, for $1 \leqslant j \leqslant n$ and $1 \leqslant i \leqslant d$. Equivalently, the $\mathcal{V}$-embedding can be defined via the transformation $x \mapsto Mx$, where $M = M(\Gamma)$ is the top left $d \times d$ submatrix of $S(P)$ and $x \in \mathbb{R}^d$ is a point in the $\mathcal{H}$-embedding. We stick to this convention for the rest of the paper. The next lemma summarizes the discussion.

Lemma 2. *Let P be a 2-level d-polytope and let $(F_1, \ldots, F_{d+1}; v_1, \ldots, v_{d+1})$ be a simplicial core for P. In the corresponding $\mathcal{H}$-embedding, all the facets of P are of the form $x(E) \leqslant 1$ or $x(E) \geqslant 0$ for some nonempty $E \subseteq [d]$. Moreover, in the corresponding $\mathcal{V}$-embedding i-th coordinate is the slack with respect to facet F_i. In particular, in the $\mathcal{V}$-embedding, all the vertices of P have $0/1$-coordinates.*

We call the submatrix $M := M(\Gamma)$ of $S(P)$ the *embedding matrix* of Γ. Note that every embedding matrix M is unimodular. Indeed, M is an invertible, lower-triangular, 0/1-matrix. Thus $\det(M) = 1$. The next lemma is the key to our approach.

Lemma 3. *In the $\mathcal{H}$-embedding P of a 2-level d-polytope with respect to any simplicial core Γ, the vertex set of P equals $P \cap M^{-1} \cdot \{0,1\}^d \subseteq \mathbb{Z}^d$, where $M = M(\Gamma)$ is the embedding matrix of Γ.*

For a hypergraph $H = (V, \mathcal{E})$ with $V = [d]$, let $P(H) := \{x \in \mathbb{R}^d \mid 0 \leqslant x(E) \leqslant 1 \text{ for each } E \in \mathcal{E}\}$. We refer to a pair of inequalities $0 \leqslant x(E) \leqslant 1$ as a pair of *hyperedge constraints* where E is a *hyperedge*. It follows from Lemma 2 that any $\mathcal{H}$-embedding of a 2-level d-polytope is of the form $P(H)$ for some hypergraph H such that $P(H)$ is integral. Conversely, each $P(H)$ that is integral is a 2-level polytope.

4 Algorithm

4.1 Closed Sets

An operator $\mathrm{cl} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$ over a ground set $\mathcal{X}$ is a *closure operator* if it is idempotent, $\mathrm{cl}(\mathrm{cl}(A)) = \mathrm{cl}(A)$; extensive, $A \subseteq \mathrm{cl}(A)$; and monotone, $A \subseteq B \implies \mathrm{cl}(A) \subseteq \mathrm{cl}(B)$. A set $A \subseteq \mathcal{X}$ is said to be *closed* with respect to cl if $\mathrm{cl}(A) = A$. In [6], Ganter and Reuter provided a polynomial delay algorithm for enumerating all the closed sets of a given closure operator.

Below, the ground set $\mathcal{X}$ will be a finite subset of points in $\mathbb{R}^d$. Let $\mathcal{F} \subseteq \mathbb{R}^d$ be another finite set of points that is disjoint from $\mathcal{X}$. For $A \subseteq \mathcal{X}$, define $\mathcal{E}_{\mathcal{F}}(A)$ to be the set of all hyperedges whose pair of hyperedge constraints is verified by $A \cup \mathcal{F}$:

$$\mathcal{E}_{\mathcal{F}}(A) := \{E \subseteq [d] \mid 0 \leqslant x(E) \leqslant 1 \text{ for every } x \in A \cup \mathcal{F}\}\,.$$

Our first closure operator is parametrized by $(\mathcal{X}, \mathcal{F})$ and is defined as:

$$\mathrm{cl}_{(\mathcal{X},\mathcal{F})}(A) := \{x \in \mathcal{X} \mid 0 \leqslant x(E) \leqslant 1 \text{ for every } E \in \mathcal{E}_{\mathcal{F}}(A)\}$$

for $A \subseteq \mathcal{X}$. In other words, $\mathrm{cl}_{(\mathcal{X},\mathcal{F})}(A)$ is the subset of $\mathcal{X}$ verifying all hyperedge inequalities that are satisfied by $A \cup \mathcal{F}$.

To obtain a 2-level d-polytope P, we fix one of its possible facets, i.e. we choose a 2-level $(d-1)$-polytope P_0 and an embedding matrix M_{d-1} of P_0. Afterwards, we extend M_{d-1} to an embedding matrix M_d of P so that P_0 is embedded in $\{x \in \mathbb{R}^d \mid x_1 = 0\} \simeq \mathbb{R}^{d-1}$ via the corresponding $\mathcal{H}$-embedding.

Then the algorithm enumerates all 2-level d-polytopes P such that M_d is an embedding matrix and P_0 is the facet defined by $x_1 \geqslant 0$ in the $\mathcal{H}$-embedding of P.

A first insight to achieve this goal is that $A := \mathrm{vert}(P) \cap \mathcal{X}$ is closed with respect to $\mathrm{cl}_{(\mathcal{X},\mathcal{F})}$, where $\mathcal{X} := \left(M_d^{-1} \cdot (\{1\} \times \{0,1\}^{d-1})\right) \setminus \{e_1\}$ and $\mathcal{F} := \mathrm{vert}(P_0) \cup \{e_1\}$. Hence, to enumerate the possible 2-level d-polytopes P with a prescribed facet P_0 and embedding matrix M_d, it suffices to enumerate the closed sets $A \subseteq \mathcal{X}$ with respect to $\mathrm{cl}_{(\mathcal{X},\mathcal{F})}$.

A second insight is that the closure operator $\mathrm{cl}_{(\mathcal{X},\mathcal{F})}$ can be improved by recalling that each facet of $P_0 \subseteq \{x \in \mathbb{R}^d \mid x_1 = 0\}$ extends uniquely to a facet of P distinct from P_0. Since each facet of P should satisfy the 2-level property, certain choices of pairs of points of $\mathcal{X}$ are forbidden. To model this, we introduce an *incompatibility graph* $G = G(P_0, M_d)$ on $\mathcal{X}$. We declare two points $u, v \in \mathcal{X}$ *incompatible* whenever there exists a facet F_0 of P_0 such that u, v and e_1 lie on three different translates of $\mathrm{aff}(F_0)$. The nodes $u, v \in \mathcal{X}$ of G are connected by an edge if and only if they are incompatible.

Next, we define the closure operator cl_G on $\mathcal{X}$ such that, for every $A \subseteq \mathcal{X}$, $\mathrm{cl}_G(A) := A$ if A is a stable set in G and $\mathrm{cl}_G(A) := \mathcal{X}$ otherwise. It can be easily checked that the composed operator $\mathrm{cl}_G \circ \mathrm{cl}_{(\mathcal{X},\mathcal{F})}$ is a closure operator over $\mathcal{X}$. This is the closure operator that we use in our enumeration algorithm.

4.2 The Enumeration Algorithm

We now provide a detailed description of our algorithm. We start with the list L_{d-1} of combinatorial types of 2-level $(d-1)$-polytopes. Each combinatorial type is stored as a slack matrix together with a simplicial core. As before, we may assume that the simplicial core is formed by the facets and vertices indexing the first $(d-1)+1$ rows and columns of the slack matrix, respectively. The algorithm below then generates the list L_d of all combinatorial types of 2-level d-polytopes, each with a simplicial core.

Theorem 1. *Algorithm 1 outputs the list of all combinatorial types of 2-level d-polytopes, each with a simplicial core.*

Proof. Consider a 2-level d-polytope P. In the rest of the proof, we consider P only as a combinatorial structure. Later on, P will be embedded in $\mathbb{R}^d$ via a $\mathcal{H}$-embedding. To simplify notation, we use the same letters for both the abstract polytope P and its realization in $\mathbb{R}^d$. We use also this convention for facets of P. We prove that a $\mathcal{H}$-embedding of P is obtained at some point by the algorithm and is added to the list L_d.

Let P_0 be any facet of P. Thus P_0 is a 2-level $(d-1)$-polytope, and hence P_0 is stored in L_{d-1} together with a simplicial core $\Gamma_0 := (F'_2, \ldots, F'_{d+1}; v_2, \ldots, v_{d+1})$. Extend Γ_0 to a simplicial core $\Gamma = (F_1, \ldots, F_{d+1}; v_1, \ldots, v_{d+1})$ for P by defining v_1 to be a vertex of F_2 not contained in F_1, and defining F_1 to be P_0 and F_i for $2 \leqslant i \leqslant d+1$ to be a unique facet of P such that $F'_i = F_i \cap P_0$. Observe that the embedding matrix $M_d := M(\Gamma)$ is of the form (1) for some $b = (b_1, \ldots, b_{d-2}) \in \{0,1\}^{d-2}$ and for $M_{d-1} := M(\Gamma_0)$.

Algorithm 1: Enumeration algorithm

Set $L_d := \varnothing$;
foreach $P_0 \in L_{d-1}$ *with simplicial core* $\mathit{\Gamma}_0 := (F'_2, \ldots, F'_{d+1}; v_2, \ldots, v_{d+1})$ **do**
    Construct the $\mathcal{H}$-embedding of P_0 in $\{0\} \times \mathbb{R}^{d-1} \simeq \mathbb{R}^{d-1}$ w.r.t. $\mathit{\Gamma}_0$;
    Let $M_{d-1} := M(\mathit{\Gamma}_0)$;
    foreach *bit vector* $b \in \{0,1\}^{d-2}$ **do**
        Complete M_{d-1} to a $d \times d$ matrix in the following way:

$$M_d := \begin{pmatrix} 1 & 0 \cdots 0 \\ 0 & \\ b_1 & \\ \vdots & M_{d-1} \\ b_{d-2} & \end{pmatrix} \tag{1}$$

        Let $\mathcal{F} := \mathrm{vert}(P_0) \cup \{e_1\}$ and $\mathcal{X} := M_d^{-1} \cdot (\{1\} \times \{0,1\}^{d-1}) \setminus \{e_1\}$;
        Let G be the incompatibility graph on $\mathcal{X}$ w.r.t. P_0 and M_d;
        Using the Ganter-Reuter algorithm [6], compute the list $\mathcal{A}$ of closed sets of the closure operator $\mathrm{cl}_G \circ \mathrm{cl}_{(\mathcal{X},\mathcal{F})}$;
        foreach $A \in \mathcal{A}$ **do**
            Let $P := \mathrm{conv}(A \cup \mathcal{F})$;
            if P *is* 2*-level and not isomorphic to any polytope in* L_d **then**
                Let $F_1 := P_0$ and $v_1 := e_1$;
                for $i = 2, \ldots, d+1$ **do**
                    Let F_i be the facet of P distinct from F_1 s.t. $F_i \supseteq F'_i$;
                end
                Add P to L_d with $\mathit{\Gamma} := (F_1, \ldots, F_{d+1}; v_1, \ldots, v_{d+1})$;
            end
        end
    end
end

Now, consider the $\mathcal{H}$-embedding of P defined by $\mathit{\Gamma}$. The vertices $v_2, \ldots, v_{d+1}$ are mapped to $e_2, \ldots, e_d$ and the origin, and v_1 is mapped to e_1. In this realization of P, the facet P_0 is embedded in $\{x \in \mathbb{R}^d \mid x_1 = 0\}$. In fact, P_0 is the facet of P defined by $x_1 \geqslant 0$.

As in the algorithm, take $\mathcal{F} := \mathrm{vert}(P_0) \cup \{e_1\}$ and $\mathcal{X} := M_d^{-1} \cdot (\{1\} \times \{0,1\}^{d-1}) \setminus \{e_1\}$. Let $A := \mathrm{vert}(P) \setminus (\mathrm{vert}(P_0) \cup \{e_1\})$. We claim that A is closed for $\mathrm{cl}_{(\mathcal{X},\mathcal{F})}$.

By Lemma 3, $A = \mathrm{vert}(P) \cap \mathcal{X}$, thus $A \subseteq \mathcal{X}$. By Lemma 2, P can be described by the linear system $\{x \in \mathbb{R}^d \mid 0 \leqslant x(E) \leqslant 1 \text{ for every } E \in \mathcal{E}_{\mathcal{F}}(A)\}$. Hence $\mathrm{vert}(P) = \mathcal{F} \cup \{x \in \mathcal{X} \mid 0 \leqslant x(E) \leqslant 1 \text{ for every } E \in \mathcal{E}_{\mathcal{F}}(A)\}$. Since $A = \mathrm{vert}(P) \cap \mathcal{X}$ and $\mathcal{X} \cap \mathcal{F} = \varnothing$, we see that $A = \mathrm{cl}_{(\mathcal{X},\mathcal{F})}(A)$. This proves the claim.

Finally, consider the incompatibility graph $G = G(P_0, M_d)$. If A were not a stable set of G then, among the facets of P adjacent to P_0, there would exist a facet that violates the 2-level property. Thus $\mathrm{cl}_G(\mathrm{cl}_{(\mathcal{X},\mathcal{F})}(A)) = A$, i.e. A is closed also with respect to $\mathrm{cl}_G \circ \mathrm{cl}_{(\mathcal{X},\mathcal{F})}$. It follows that the combinatorial type of P is added at some point by the algorithm to the list L_d.

Clearly, L_d contains at most one member for each combinatorial type of 2-level d-polytope, because a 2-level polytope is added to L_d only if it is not isomorphic to any other polytope in the list. □

4.3 Implementation

We implement the algorithm presented in Section 4.2 in `Perl`. We use `polymake` [7] for the geometric computations, such as congruence and isomorphism tests, convex hull and f-vector computations, and general linear algebra operations.

Isomorphism testing is in general a harder problem than congruence testing for polytopes given by sets of vertices, as it involves a convex hull computation. For this reason, before computing the convex hull in Step 11, we first ascertain whether or not there is an existing congruent polytope in L_d by testing the corresponding sets of vertices. For congruence tests, `polymake` uses the reduction of the congruence problem for arbitrary point sets to the graph isomorphism problem [2]. For isomorphism tests, the problem is reduced to graph isomorphism of the vertex-facet incidence graphs. For the 2-level test in Step 12 we check if every facet inequality of P computed by a convex hull algorithm in Step 11 attains two values when evaluated on vertices of P.

As part of our code, we implement the Ganter-Reuter algorithm [6]. The sets are represented by bit vectors and all the operations we need—such as order test between two sets, and closed set computations—are implemented by bit operations. For these we use the `Perl` library `Bit::Vector` [3].

The choice of the closed sets enumeration algorithm is not crucial for our problem: experiments indicate that more than 99% of the enumeration time is spent in geometric computation (i.e. convex hull and isomorphism tests) and the rest is spent computing the next closed set from the current one.

Since convex hull computation is crucial for our enumeration algorithm, we perform experiments on the performance of 4 state-of-the-art convex hull implementations: `beneath_beyond (bb)`, which implements the incremental beneath and beyond algorithm; `lrs`, which implements the reverse search algorithm; and `cdd`, `ppl`, which implement the double description method. In $d = 6$ without redundancy removal the fastest implementation is `bb`, `cdd`, `lrs`, `ppl` for 224, 23, 3, 879 polytopes respectively and with redundancy removal for 28, 45, 376,

Table 1. Numbers of non-isomorphic 2-level polytopes, equivalence classes (isomorphic and congruent) and closed sets computed by the algorithm.

d	closed sets	2-level	isomorphic	congruent	closed sets/2-level
4	277	19	203	42	0.95
5	10963	106	7669	621	0.77
6	1908641	1150	414714	42076	0.24

Table 2. Numbers of combinatorially equivalent 0/1 polytopes, 2-level polytopes and sub-classes; 2L: 2-level polytopes, Δ-f: with one simplicial facet, STAB: stable sets of perfect graphs, polar: 2-level polytopes whose polar is 2-level, CS: centrally symmetric, Birk: Birkhoff polytope faces from [16], '-': exact numbers are unknown.

d	2L	Δ-f	STAB	polar	CS	Birk	0/1
3	5	4	4	4	2	4	8
4	19	12	11	12	4	11	192
5	106	41	33	42	13	33	1,048,576
6	1150	248	148	276	45	129	-
7	-	-	-	-	238	661	-

700 polytopes respectively. Interestingly, `lrs` and `bb` exchange roles in these two cases. We conclude that `ppl` is the most efficient implementation in most of the cases and thus the one we choose for our implementation. Note that since we know that the input points are always in convex position we can avoid redundancy removal thus earn a $5\times$ speed-up in dimension 6.

5 Experimental Results

5.1 Outcome of the Experiments

In dimension 4, the set of 2-level polytopes is computed by our algorithm in 20 seconds, while for $d = 5$ it takes 12 minutes to enumerate 106 2-level polytopes on an `Intel(R) Core(TM) i7-4700HQ CPU @ 2.40GHz`. For $d = 6$ we exploited one property of our algorithm: its straightforward parallelization. We created one job for each branch of commands in the two outer for loops of the algorithm and submitted these jobs to a cluster[1]. In particular, we created one job for each 2-level 5-polytope and each b vector, i.e. 1696 jobs. The total computation lasted 1 day (the sequential time is estimated in 4.5 days).

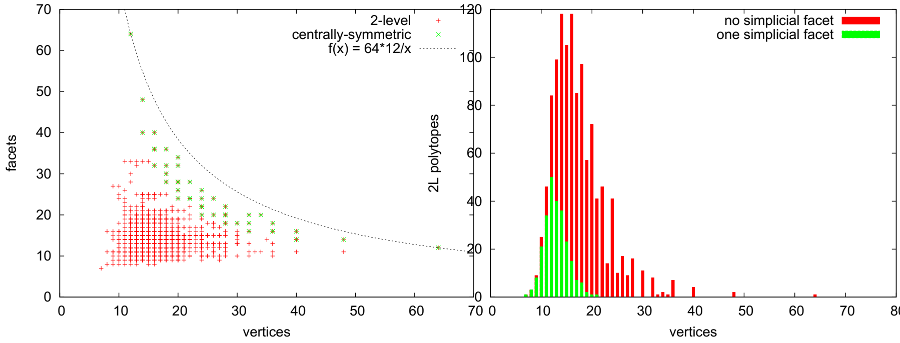
We illustrate the attained speed-up gained by using $\mathrm{cl}_G \circ \mathrm{cl}_{(\mathcal{X},\mathcal{F})}$ instead of $\mathrm{cl}_{(\mathcal{X},\mathcal{F})}$. In $d = 6$ the use of the first leads to $\sim 1.9 \cdot 10^6$ closed sets while the later to $\sim 10^8$ closed sets.

Not surprisingly, as the dimension increases, more computation time is consumed in testing polytopes that are not 2-level as depicted in Table 1.

To understand the current limits of computation note that in $d = 7$ we have to create 36800 jobs, while experiments show there are jobs that need more than 5 days to terminate.

Table 2 summarizes our results regarding the number of 2-level polytopes and interesting subclasses. Our main result is the number of isomorphism classes of 2-level polytopes in $d = 6$. Additionally, we make use of properties of 2-level centrally symmetric polytopes to enumerate all of them in $d = 7$.

[1] Hydra balanced cluster: `https://cc.ulb.ac.be/hpc/hydra.php`



(a) (b)

Fig. 1. (a) The relation between the number of facets and the number of vertices of 2-level 6-polytopes; (b) the number of 2-level 6-polytopes and the class with the ones with a simplicial facet as a function of the number of vertices.

The computed polytopes in `polymake` format as well as more information on the experiments and data are available online[2]. Taking advantage of the computed data we perform a number of statistical tests to understand the structure and properties of 2-level polytopes.

We experimentally study the number of 2-level polytopes as a function of the number of vertices in dimension 6 (see Fig. 1(b)). Interestingly, most of the polytopes, namely 1048 (i.e. more than 90%) have 10 to 24 vertices. The number of polytopes with a simplicial facet is maximum when the number of vertices is 12 and the extreme cases are the simplex (7 vertices) and the hypersimplex $\Delta_6(2)$ (21 vertices) [21].

The relation between the number of vertices and the number of facets in $d = 6$ is depicted in Fig. 1(a). Experiments show that the bound $f_0(P)f_{d-1}(P) \leqslant d2^{d+1}$ holds for all 2-level d-polytopes up to $d = 6$ and for the centrally symmetric 2-level polytopes in $d = 7$. Note that $f_0(P)f_{d-1}(P) = d2^{d+1}$ when P is the cube or its polar.

Our experiments show that all 2-level centrally symmetric polytopes, up to dimension 7, validate Kalai's 3^d conjecture [14] (note that for general centrally symmetric polytopes, Kalai's conjecture is known to be true only up to dimension 4 [17]). Dimension 5 is the lowest dimension in which we found centrally symmetric polytopes that are not Hanner nor Hansen (e.g. one with f-vector $(12, 60, 120, 90, 20)$). In dimension 6 we found a 2-level centrally symmetric polytope with f-vector $(20, 120, 290, 310, 144, 24)$, for which therefore $f_0 + f_4 = 44$; this offers a stronger counterexample to the conjecture B of [14] than the one presented in [17] having $f_0 + f_4 = 48$.

---

[2] `http://homepages.ulb.ac.be/~vfisikop/data/2-level.html`
Note that the stored polytopes are in a slightly different format than described in the algorithm, i.e. we store an $\mathcal{H}$-embedding without the slack matrix and the simplicial core.

6 Discussion

We think that the experimental evidence we gathered will lead to interesting research questions. As a sample, we propose the following question motivated by Fig. 1(a): is it true that $f_0(P)f_{d-1}(P) \leqslant d2^{d+1}$ for all 2-level d-polytopes P? And if yes, is equality attained only by the cube and cross-polytope? It is known that $f_0(P) \leqslant 2^d$ with equality if and only if P is a cube and $f_{d-1}(P) \leqslant 2^d$ with equality if and only if P is a cross-polytope [8]. In these cases, $f_0(P)f_{d-1}(P) = d2^{d+1}$.

One way to decrease the computation time of the algorithm is to exploit the symmetries of the embedding matrix M_d and reduce the possible choices for the bit vector b. Given M_{d-1} two vectors b are equivalent if the resulting matrices M_d can be transformed from one to the other by swapping columns and rows. Therefore, only one b for each equivalent class should be considered by the algorithm.

Acknowledgments. We acknowledge support from the following research grants: ERC grant *FOREFRONT* (grant agreement no. 615640) funded by the European Research Council under the EU's 7th Framework Programme (FP7/2007-2013), *Ambizione* grant PZ00P2 154779 *Tight formulations of 0-1 problems* funded by the Swiss National Science Foundation, the research grant *Semidefinite extended formulations* (Semaphore 14620017) funded by F.R.S.-FNRS, and the *ARC* grant AUWB-2012-12/17-ULB2 *COPHYMA* funded by the French community of Belgium. We also thank the anonymous referees for their comments which helped us improve the presentation.

References

1. Aichholzer, O.: Extremal properties of 0/1-polytopes of dimension 5. In: Ziegler, G., Kalai, G. (eds.) Polytopes - Combinatorics and Computation, pp. 111–130. Birkhäuser (2000)
2. Akutsu, T.: On determining the congruence of point sets in d dimensions. Computational Geometry 9(4), 247–256 (1998)
3. Beyer, S.: Comprehensive perl archive network: Bit-vector-7.4 (2014). `http://search.cpan.org/~stbey/Bit-Vector-7.4`
4. Chvátal, V.: On certain polytopes associated with graphs. J. Combinatorial Theory Ser. B 18, 138–154 (1975)
5. De Loera, J., Rambau, J., Santos, F.: Triangulations. Algorithms and Computation in Mathematics, vol. 25 (2010)
6. Ganter, B., Reuter, K.: Finding all closed sets: A general approach. Order 8(3), 283–290 (1991) (English)
7. Gawrilow, E., Joswig, M.: Polymake: an approach to modular software design in computational geometry. In: International Symposium on Computational Geometry (SOCG), pp. 222–231. ACM Press (2001)
8. Gouveia, J., Parrilo, P., Thomas, R.: Theta bodies for polynomial ideals. SIAM Journal on Optimization 20(4), 2097–2118 (2010)
9. Gouveia, J., Robinson, R., Thomas, R.: Polytopes of minimum positive semidefinite rank. Discrete & Computational Geometry 50(3), 679–699 (2013)

10. Grande, F., Sanyal, R.: Theta rank, levelness, and matroid minors, arXiv preprint, arXiv:1408.1262 (2014)
11. Haase, C.: Lattice polytopes and unimodular triangulations, Ph.D. thesis, Technical University of Berlin (2000)
12. Hanner, O.: Intersections of translates of convex bodies. Mathematica Scandinavica 4, 65–87 (1956)
13. Hansen, A.: On a certain class of polytopes associated with independence systems. Mathematica Scandinavica 41, 225–241 (1977)
14. Kalai, G.: The number of faces of centrally-symmetric polytopes. Graphs and Combinatorics 5(1), 389–391 (1989)
15. Kuznetsov, S., Obiedkov, S.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental and Theoretical Artificial Intelligence 14, 189–216 (2002)
16. Paffenholz, A.: Faces of Birkhoff polytopes. Electronic Journal of Combinatorics 22 (2015)
17. Sanyal, R., Werner, A., Ziegler, G.: On Kalai's conjectures concerning centrally symmetric polytopes. Discrete & Computational Geometry 41(2), 183–198 (2009)
18. Stanley, R.: Decompositions of rational convex polytopes. Annals of Discrete Mathematics, 333–342 (1980)
19. Stanley, R.: Two poset polytopes. Discrete & Computational Geometry 1(1), 9–23 (1986)
20. Sullivant, S.: Compressed polytopes and statistical disclosure limitation. Tohoku Mathematical Journal, Second Series 58(3), 433–445 (2006)
21. Ziegler, G.: Lectures on polytopes, vol. 152, Springer Science & Business Media (1995)
22. Ziegler, G.: Lectures on 0/1-polytopes. In: Kalai, G., Ziegler, G. (eds.) Polytopes - Combinatorics and Computation. DMV Seminar, vol. 29, pp. 1–41. Springer, Birkhäuser, Basel (2000)