Finding a Simple Polytope from its Graph in Polynomial Time

Eric J. Friedman

School of Operations Research and Information Engineering, Cornell University
ejf27@cornell.edu,
http://www.people.cornell.edu/pages/ejf27/

Abstract. We show that one can compute the combinatorial facets of a simple polytope from its graph in polynomial time. Our proof relies on a primal-dual characterization (by Joswig, Kaibel and Korner) and a linear program, with an exponential number of constraints, which can be used to construct the solution and can be solved in polynomial time. We show that this allows one to characterize the face lattice of the polytope, via a simple face recognition algorithm. In addition, we define the concept of a pseudo-polytopal-multigraph which may be of independent interest.

1 Introduction

In [1], Blind and Mani proved, using tools from homology theory, a conjecture of Perles [5] – that one can construct the entire face lattice of a simple[1] polytope from its graph. Then Kalai [5] presented an elegant elementary proof of this result. Whereas Blind and Mani's result was essentially nonconstructive, Kalai's result was constructive but required exponential time to compute (in the size of the graph).

More recently, Joswig, Kaibel and Korner [2] extended Kalai's analysis to provide polynomial certificates for this problem, based on a pair of combinatorial optimization problems that form a primal dual pair. However, they did not provide polynomial algorithms for either of these problems and thus left open the question of whether this problem can be solved in polynomial time.

In this paper, we present a polynomial time algorithm for computing the facets of a polytope from its graph, resolving this question. We present a linear program for computing the 2-faces of the polytope from its graph which can be solved in polynomial time. As discussed in [2] this resolves the issue, as it is straightforward to compute the facets of the polytope from the set of 2-faces, in polynomial time. Also, as we show, it is straightforward to characterize the face lattice of the polytope, via a simple face recognition algorithm, i.e., there is a

[1] Note that if the polytope is not simple then it is not uniquely defined by its graph. Thus, we will only consider simple polytopes.

polynomial algorithm for deciding whether a set of vertices form a face of the polytope.[2]

Our discussion in the remainder of the paper will be self contained, but compact. See the book [10] for more details on the geometry of polytopes, [2] for more technical details on some of the specific techniques we use and [3] for an excellent review, including the history, of the problem studied in this paper.

2 Polytopes and Graphs

In this paper we will be concerned with the "combinatorial structure" of a polytope. Thus, we are not concerned with actual realization of the polytope in Euclidean space; however the existence of such an embedding is necessary for the analysis. The following definitions are all standard and are discussed in more detail in standard texts, e.g. [10].

Given a simple[3] polytope, $P \subset \Re^d$, with vertices V, we consider its face lattice whose elements are the subsets of V which correspond to faces, where $V_k(P)$ is the set of k-faces of P. For example, the elements of $V = V_0$ are all 0-faces, while the sets of vertices that form facets, V_{d-1}, are $d-1$-faces. The face lattice is ordered by inclusion.

The graph of the polytope, $G(P)$, is the undirected graph formed by the vertices and their edges (or 1-faces). An orientation on a graph corresponds to a directed version on this graph, and an acyclic orientation is one in which there are no directed cycles.

An abstract objective function (AOF) on $G(P)$ is an acyclic orientation such that the induced directed subgraph on every face of P has a unique sink.

3 2-systems and Pseudo-Polytopes

Now we describe a key lemma. Let $G = (V, E)$ be the graph of a simple polytope, P, in $\Re^d$, where V is the set of vertices of the polytope and E are its edges.

A 2-frame, centered at v, is a set of three distinct nodes, (v, v', v'') such that $(v, v') \in E$ and $(v, v'') \in E$. A 2-system is a set of cycles in G such that every 2-frame is contained in a unique cycle.

Let $\mathcal{O}$ be an acyclic orientation on G. Define $\mathcal{H}(\mathcal{O})$ to be the number of 2-frames that are sinks under $\mathcal{O}$, where the 2-frame (v, v', v'') is a sink if both edges (v, v') and (v, v'') are oriented towards the center of the frame, v.

Our analysis will be based on the following (minor) extension of the main result from [2]. Our modification is that we require a specified vertex not be a source.

---

[2] In [4] an algorithm is presented for computing the entire face lattice in a time that is linear in the size of the face lattice. However, this may be exponential in the size of the initial graph, as there can be exponentially many faces.

[3] A simple polytope in $\Re^d$ is one for which every vertex is adjacent to exactly d edges.

Theorem 1 (Joswig, Kaibel, and Korner). *Let P be a simple d-polytope. For every 2-system S of G, vertex $v \in V$, and every acyclic orientation $\mathcal{O}$ of G,* such that v is a source*, the inequalities*

$$|S| \leq |V_2(P)| \leq \mathcal{H}(\mathcal{O})$$

hold, where the the second holds with equality if and only if $\mathcal{O}$ is an AOF.

Proof: Our proof is a slight modification of that in [2], since we require that a chosen vertex be a source. First note that for an acyclic orientation every cycle must contain a 2-sink. Thus we must have that $|S| \leq \mathcal{H}(\mathcal{O})$ and $|V_2(P)| \leq \mathcal{H}(\mathcal{O})$ since $V_2(P)$ is a 2-system. Next note that if $\mathcal{O}$ is an AOF then $|V_2(P)| = \mathcal{H}(\mathcal{O})$ by definition. Note that such an AOF exists since there exists a linear function on P where v is a minimum and linear functions on polytopes generate AOFs as long as no two vertices have the same value. One can construct such an AOF by taking a small perturbation of a linear function for which a supporting hyperplane of v is a level set.

Combining these inequalities completes the proof: $|S| \leq |V_2(P)|$ and $|V_2(P)| \leq \mathcal{H}(\mathcal{O})$. □

Note that Joswig, Kaibel, and Korner [2] also show that if $|S| = |V_2(P)|$ then $S = V_2(P)$. Their proof still holds in our setting, but we don't include it here as it will follow directly from our analysis.

Thus, if we can find a 2-system S that uniquely maximizes $|S|$ in polynomial time, then we have found $V_2(p)$ and from that it is straightforward to compute facets of P in polynomial time. See [2, 3] for details.

4 Pseudo-Polytopal Multigraphs

We use the above theorem to define a "pseudo-polytopal multigraph" to be a multi-graph G such that there exists a vertex v and "pseudo 2-face set", $V_2(G)$ such that Theorem 1 holds. Clearly the graph of a polytope is pseudo-polytopal; however, as we now show, other multi-graphs (which do not arise from simple polytopes) may also be pseudo-polytopal.

Given a graph G of a polytope P define the contraction of G by a 2-face f to be a new multi-graph $C_f(G)$, where all the nodes in f are contracted to a single node, denoted v. Note that this is a multi-graph as there may be multiple edges connecting v to an adjacent node. We consider each of these to be distinct and may even have a 2-face on only 2 nodes.

Theorem 2. *Let G be the graph of a simple polytope P and f be a 2-face of P. Then $G' = C_f(G)$ is a pseudo-polytopal multigraph.*

Proof: The proof is identical to the proof of Theorem 1 where we choose $V_2(G')$ to be the $V_2(P) \setminus f$ and $\mathcal{O}$ to be the contraction of an AOF for G where the edges from all vertices on the face f point towards those that are not in f. To construct such an AOF simply take the linear objective function with f as a level set and perturb it slightly. □

5 Computing $|V_2(P)|$

In this section, we will present a binary integer program with an exponential number of variables for computing this 2-system. Somewhat surprisingly, this can be solved in polynomial time.

Let T be the set of all 2-frames in G and $t \in T$ be the 2-frame (t_0, t_{-1}, t_1) centered at t_0. Let W be the set of simple cycles in G. Then by Theorem 1 in order to compute $|V_2(P)|$ we need to solve:

$$max \sum_{w \in W} x_w \qquad (IP-S)$$
$$s.t.$$
$$\forall t \in T: \quad \sum_{w \ni t} x_w = 1$$
$$x_w \in \{0, 1\}$$

Where we write $w \ni t$ as a shorthand for the 2-frames t contained in w. First we consider the following linear relaxation of this integer program.

$$max \sum_{w \in W} x_w \quad \text{(LP-S)}$$
$$s.t.$$
$$\forall t \in T: \quad \sum_{w \ni t} x_w \leq 1$$
$$x_w \geq 1$$

Next, we consider the dual of this linear program:

$$min \sum_{t \in T} v_t \quad \text{(LP-SD)}$$
$$s.t.$$
$$\forall w \in W: \quad \sum_{t \in w} v_t \geq 1$$
$$v_t \geq 0$$

Let IP-SD be the related binary integer program for LP-SD, i.e., replace $0 \leq v_t$ with $v_t \in \{0, 1\}$.

Now, consider an acyclic orientation, $\mathcal{O}$ of G and let $v_t = 1$ represent the case when the 2-frame t is a 2-sink. Then the integer program for minimizing $\mathcal{H}(\mathcal{O})$ over all acyclic orientations can be written by adding the constraint that v must arise from an acyclic orientation on G, to IP-SD.

$$min \sum_{t \in T} v_t \quad \text{(IP-}\mathcal{H}\text{)}$$

$$s.t.$$
$$\forall w \in W : \quad \sum_{t \in w} v_t \geq 1$$
$$v_t \geq 0$$
$$v_t \text{ arises from an acyclic orientation of } G$$

The application of Theorem 1 to this sequence of optimization problems yields the following result:

Theorem 3. *Let P be a simple d-polytope with graph G. Then the following optimization problems for G all have the same optimal value: IP-S, LP-S, LP-SD, IP-SD and IP-$\mathcal{H}$.*

Proof: Let $Opt(problem)$ be the optimal objective value for the optimization problem, "problem". Then it is easy to see that $Opt(IP-S) \leq Opt(LP-S)$ and $Opt(LP-SD) \leq Opt(IP-SD) \leq Opt(IP-\mathcal{H})$ as these are sequences of relaxations. By strong duality, we have $Opt(LP-S) = Opt(LP-SD)$. Now, Theorem 1 completes the proof since it implies that $Opt(IP-S) = Opt(IP-\mathcal{H})$. □

Thus, we can compute $|V_2(P)|$ by solving either LP-S or LP-SD and in fact compute $V_2(P)$ if the optimal solution to LP-S is unique, which we will show below.

6 Solving the Linear Program

Now we show that LP-S can be solved in polynomial time using the ellipsoid algorithm. This is because the dual, LP-SD, has a polynomial number of variables and all constraints have polynomially bounded size (since all the coefficients are 0 or 1). Then, because of the equivalence of separation and optimization (see Corollary 14.1g in [8]) all we need to show is that there exists a polynomial time separation algorithm for LP-SD.

A separation algorithm in this case is an algorithm which given a vector v can check whether v is feasible and, if not, find a constraint violated by v. In our case such a constraint is a cycle $w \in W$ such that $\sum_{t \in w} w_t < 1$.

First we construct the directed graph G^T which has a vertex for every *ordered* 2-frame, e.g., if (v, v', v'') is a frame centered at v then both $\tau = (v, v', v'')$ and $\tau' = (v, v'', v')$ are distinct vertices of G^T. There is a directed edge from $\tau = (v, v', v'')$ to $\tau' = (w, w', w'')$ if $v = w'$ and $v'' = w$. If the edge from τ to τ' exists then its "cost" is given by $(v_\tau + v_{\tau'})/2$. It is easy to see that any cycle in this graph has a corresponding cycle in $G(P)$ and the total cost of that cycle is the sum of the v_t's along that cycle, which corresponds to the constraints in LP-SD.

Thus, if the minimum cost cycle in G_T has total cost less than 1 it yields a violated constraint and if it has cost greater than or equal to 1 then the current solution is feasible. Finding the minimum cost cycle can be done in polynomial time using dynamic programming. A less efficient, but still polynomial time method can be done as follows:

For every vertex v consider the graph G_v^T which is constructed from G_T by removing all 2-frames which are centered at vertex v. Next consider all pairs of vertices, (τ, τ') where $\tau = (v', v, v'')$, $\tau' = (w', w'', v)$ and $v' \neq w'$. Compute the shortest path from t to t' and add the cost of the 2-frame (v, w', v'). The minimum over all such pairs yields the minimum cost cycle that includes vertex v. Repeating this procedure for every vertex, allows one to compute the minimum cost cycle for G_T.

Thus, we have shown:

Theorem 4. *LP-S is solvable in polynomial time.*

7 Uniqueness of the Optimal Solution

To complete the analysis one must guarantee that the solution of LP-S is unique.

Theorem 5. *LP-S has a unique optimal solution.*

Proof: Consider an optimal solution of IP-S, x^* and note that it is also an optimal extreme point solution of LP-S by Theorem 1. Assume that there exists a second optimal extreme point solution of LP-S, $x' \neq x^*$. Then there must exist some $w \in W$ such that $x^*_w = 1$ and $x'_w = 0$, otherwise $(1+\epsilon)x' - \epsilon x^*$ would also be an optimal solution, for small enough $\epsilon > 0$, implying that x'_w is not an extreme point.

Let $f \in F$ denote the 2-face implied by x^*_w and contract the graph G by f, denoting this node by f and the contracted graph by $G' = C_f(G)$.

Now consider LP-S on this graph where we ignore the constraints for 2-frames centered at f but require all the remaining ones. Since G' is psuedo-polytopal our previous argument holds for the string of optimization problems induced by G'. In particular, the solution of IP-S must have objective value equal to $|V_2(P)| - 1$; however the projection of x' onto G' is feasible for LP-S but has a greater objective value ($|V_2(P)|$), providing a contradiction and proving the theorem. □

Note that LP-SD, IP-SD and IP-$\mathcal{H}$ all have multiple optimal solutions as a polytope has many different AOFs.

8 Main Result

Thus, a solution of LP-S, which we can compute in polynomial time, is also the unique optimal solution of IP-S, which yields the set of 2-faces for P. Thus, we have proven our main result.

Theorem 6. *One can compute the set of combinatorial 2-faces of a polytope from its graph in time that is polynomial in the size of the graph.*

To construct the facets, one needs simply to "grow" them from the 2-faces. To grow a facet, start with any (d-1)-frame centered at v and let v' be the unique vertex adjacent to v which is not in that frame. Consider any other vertex, v'', in that frame, which is not the center of the frame. Clearly there exists another frame in the same facet as the first (d-1)-frame which is centered at v''. Now consider the vertex $\hat{v}$ adjacent to v'' which is contained in the 2-face that contains the frame (v, v', v''). It is straightforward to show that the (d-1)-frame centered at v'' in that facet contains all the vertices except $\hat{v}$ ([3]). Clearly one can use this procedure to find all the facets in polynomial time.

Corollary 1. *One can compute the set of combinatorial facets of a polytope from its graph in time that is polynomial in the size of the graph.*

We note that the face lattice may contain exponentially many faces, so explicitly exhibiting it may require exponential time. However, on can implicitly compute the face lattice in polynomial time.

Corollary 2. *Let $G = (V, E)$ be the graph of a polytope, P, and $W \subset V$. Then one can compute whether W is a (combinatorial) face of P in polynomial time.*

Proof: From the previous corollary, one can compute the facets of P in polynomial time. It is easy to see that W is a face of P if and only if there exists a set of facets of P, $F = \{F_1, \ldots, F_k\}$ such that $W = \bigcap_{i=1\ldots k} F_i$. This can be computed in polynomial time by letting F be the set of all facets which contain W. $\square$

9 Non-Simple Polytopes

It is tempting to extend our analysis to non-simple graphs. For example, given a graph of a non-simple polytope, G, can one find *some* polytope which has G as its graph. The key impediment in the analysis is the determination of the 2-frames. While every triplet of vertices v, v', v'' such that (v, v') and (v, v'') are both elements of E is a 2-frame in a simple polytope, this is not true in general. For example, when $d = 3$, a vertex of degree 4 should only have 4 2-frames, not 6. Thus, prior to solving LP-S one needs to choose a consistent set of 2-frames. If the choice of 2-frames is correct, then our analysis extends and one can compute a set of 2-faces (and hence a set of facets) that correspond to some realization polytope.

For a graph that is "mostly simple", that is a graph for which all but a fixed number of vertices are of degree d one could compute the "2-faces", by solving LP-S, for every consistent set of 2-frames. However, finding a correct set of 2-faces is problematic as it is not necessarily true that the minimal solution leads to a set of 2 faces for a true polytope, although this is a reasonable conjecture.

10 AOFs and USOs

Our analysis suggests that similar techniques might be useful for finding abstract objective functions in polynomial time, an interesting open problem. In addition, these techniques also might be useful for understanding unique sink orientations [9] which drops the requirement that the orientation be acyclic in the integer program IP-$\mathcal{H}$. Also note that IP-SD suggests that a generalization of unique sink orientations which drops the requirement that the sinks arise from an orientation, might also be of interest. The relevant parts of Theorem 1 are valid for all of these relaxations of AOFs.

11 Polynomially Computable Classes of Almost-Polytopal Graphs

It is known that the problem of checking whether a graph is polytopal is not in co-NP and therefore not in P as discussed in [3]. (In fact, universality theorems [6, 7], suggest (but do not prove) that it may even be NP-hard or Pspace-hard.) Thus, it is interesting to find efficiently computable subclasses of graphs which contain the graphs of simple polytopes. Let $\mathcal{G}$ be the set of all graphs and $\mathcal{G}^{SP}$ be the set of all graphs which arise from simple polytopes.

For example, define $\mathcal{G}^{RC}$ to be the class of graphs which for some d are d-regular and d-connected. Then clearly membership in $\mathcal{G}^{RC}$ is polynomial computable and $\mathcal{G}^{SP} \subset \mathcal{G}^{RC} \subset \mathcal{G}$, where both inclusions are strict. We are not aware of any significantly tighter classes of graphs in the literature for which membership is polynomial computable.

However, our main lemma provides some new ones. Let $\mathcal{G}^{PD}$ be the subset of $\mathcal{G}^{RC}$ for which the linear program (LP-S) has a unique integral solution. One can tighten this class to $\mathcal{G}^{PDF}$ by using the solution (a set of "2-faces") to construct the facets of the graph using the procedure discussed above and then checking that this set of "facets" is "consistent", e.g., every $(d-1)$-frame is covered exactly once.

Clearly membership in $\mathcal{G}^{PD}$ and $\mathcal{G}^{PDF}$ can be computed in polynomial time and $\mathcal{G}^{SP} \subset \mathcal{G}^{PDF} \subseteq \mathcal{G}^{PD} \subseteq \mathcal{G}^{RC} \subset \mathcal{G}$, where we conjecture that all inclusions are strict. However, we do not know the utility of these new classes.

12 Acknowledgements

I'd like to thank Adrian Lewis, David Shmoys, Volker Kaibel, Mike Todd and Gunter Ziegler for helpful comments. This research has been supported in part by the NSF under grant ITR-0325453.

References

1. R. Blind and P. Mani-Levitska. On puzzles and polytope isomorphisms. *Aequationes Math.*, 34:287297, 1987.
2. M. Joswig, V. Kaibel, and F. Körner. On the k-systems of a simple polytope. *Isr. J. Math.*, 129:109–118, 2002.

3. V. Kaibel. Reconstructing a simple polytope from its graph. In M. Junger, G. Reinelt, and G. Rinaldi, editors, *Combinatorial Optimization – Eureka, You Shrink!*, pages 105–118. Springer, New York, 2002.
4. V. Kaibel and M. Pfetsch. Computing the face lattice of a polytope from its vertex-facet incidences. *Comput. Geom.*, 23:281–290, 2002.
5. G. Kalai. A simple way to tell a simple polytope from its graph. *J. Comb. Theory, Ser. A*, 49:381–383, 1988.
6. N. E. Mnev. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In *Topology and geometry—Rohlin Seminar*, volume 1346, pages 527–543. Springer, Berlin - New York, 1988.
7. J. Richter-Gebert and G. M. Ziegler. Realization spaces of 4-polytopes are universal. *Bull. Amer. Math. Soc.*, 32(4):403–412, 1995.
8. A. Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience series in discrete mathematics and optimization. John Wiley and Sons, Chichester, 2000.
9. T. Szabó and E. Welzl. Unique sink orientations of cubes. In *Proc. 42nd Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 547–555, New York, 2001.
10. G. M. Ziegler. *Lectures on Polytopes.* Springer Verlag, New York, 1998.