On the Complexity of Polytope Isomorphism Problems

Volker Kaibel[*]**, Alexander Schwartz**

Technische Universität Berlin
Fakultät II, Institut für Mathematik, MA 6–2
Straße des 17. Juni 136
10623 Berlin, Germany
e-mail: {schwartz,kaibel}@math.tu-berlin.de

July 2002

(To appear in Graphs and Combinatorics.)

Abstract. We show that the problem to decide whether two (convex) polytopes, given by their vertex-facet incidences, are combinatorially isomorphic is graph isomorphism complete, even for simple or simplicial polytopes. On the other hand, we give a polynomial time algorithm for the combinatorial polytope isomorphism problem in bounded dimensions. Furthermore, we derive that the problems to decide whether two polytopes, given either by vertex or by facet descriptions, are projectively or affinely isomorphic are graph isomorphism hard.

Key words. polytope isomorphism, equivalence of polytopes, graph isomorphism complete, graph isomorphism hard, polytope congruence

MSC 2000: 52B05, 05C60, 52B11, 68R10

1. Introduction

Combinatorial isomorphism of polytopes. When treated as a combinatorial object, a polytope (i.e., a bounded convex polyhedron) is identified with its *face lattice*, i.e., the lattice formed by its faces, which are ordered by inclusion. Two polytopes are considered *combinatorially isomorphic* if their face lattices are isomorphic, i.e., if there is an in both directions inclusion preserving bijection between their sets of faces.

[*] Supported by a DFG Gerhard-Hess-Forschungsförderungspreis donated to Günter M. Ziegler (Zi 475/2-3).

Since the face lattice is both atomic and coatomic, the entire combinatorial structure of a polytope P is encoded in its *vertex-facet incidences*, i.e., in a bipartite graph $\mathcal{I}(P)$, whose two shores represent the vertices and facets, where an edge indicates that the vertex corresponding to the one end node is contained in the facet corresponding to the other one. The *polytope isomorphism problem* is the problem to decide whether two polytopes P and Q, given by their vertex-facet incidences, are combinatorially isomorphic. For all concepts and notations concerning polytope theory, we refer to Ziegler's book [32].

Two graphs G and G' are called *isomorphic* if there is a bijection φ between their vertex sets such that $\{\varphi(v), \varphi(w)\}$ is an edge of G' if and only if $\{v, w\}$ is an edge of G. The problem to decide whether two graphs G and G' are isomorphic is known as the *graph isomorphism problem*. If the nodes and edges of the two graphs are labeled, and if we consider only isomorphisms that preserve the labels, then the corresponding decision problem is called the *labeled graph isomorphism problem*. Clearly, the (usual) graph isomorphism problem is the special case of the labeled graph isomorphism problem, where all nodes have the same label. Conversely, the labeled graph isomorphism problem can be Karp-reduced (see below) to the graph isomorphism problem.

The polytope isomorphism problem for two polytopes P and Q is a special case of the labeled graph isomorphism problem, where in the bipartite graphs $\mathcal{I}(P)$ and $\mathcal{I}(Q)$ each node is either labeled "vertex" or "facet."

Since every isomorphism between two polytopes or between two graphs is determined by its restriction to the vertices or nodes, respectively, we will often identify an isomorphism with that restriction. Thus, two polytopes are combinatorially isomorphic if there is a bijection between their sets of vertices that induces a bijection between their sets of facets.

Checking isomorphism by computer. For computer systems dealing with (the combinatorial structures of) polytopes, the problem of checking two polytopes for combinatorial isomorphism is quite important. Grünbaum writes in his classical book [12, footnote on p. 39]:

> *For two given polytopes it is, in principle, easy to determine whether they are combinatorially equivalent or not. It is enough (. . .) to check whether there exists any inclusion preserving one-to-one correspondence between the two sets of faces. However, this procedure is practically feasible only if the number of faces is rather small.*

The `polymake`-system of Gawrilow and Joswig [11] currently implements the isomorphism test for two polytopes P and Q by checking whether the bipartite graphs $\mathcal{I}(P)$ and $\mathcal{I}(Q)$ are isomorphic as labeled graphs. This is done by using the software package `nauty` by McKay [25].

One of our results (Theorem 2) shows that in order to solve the general polytope isomorphism problem the only way is indeed to use an algorithm for the general graph isomorphism problem. Our second main result (Theorem 4) shows that one might take advantage of the fact that the polytopes dealt with in computer systems usually have rather small dimensions.

Geometric notions of isomorphism of polytopes. If one is concerned with coordinate representations of polytopes then geometric notions of isomorphism become important. In particular, two polytopes $P \subset \mathbb{R}^{d'}$, $Q \subset \mathbb{R}^{d''}$ are called *affinely* or *projectively isomorphic* if there is an affine or projective, respectively, map from $\mathbb{R}^{d'}$ to $\mathbb{R}^{d''}$ inducing a bijection between P to Q. Two polytopes are *congruent* if there is an affine isomorphism between them that is induced by an orthogonal matrix.

The decision problems corresponding to these geometric notions of isomorphism are the *polytope congruence problem* and the *affine* or *projective polytope isomorphism problem*, where for each of them the two polytopes either are given by vertex coordinates ($\mathcal{V}$-descriptions) or by inequality coefficients ($\mathcal{H}$-descriptions).

The chain of implications "congruent $\Rightarrow$ affinely isomorphic $\Rightarrow$ projectively isomorphic $\Rightarrow$ combinatorially isomorphic" is well-known. We will mainly be concerned with combinatorial isomorphism; nevertheless, our results also have implications for geometric isomorphism (see Theorem 3).

Graph isomorphism and reductions. The complexity status of the general graph isomorphism problem is open. While it is obvious that the problem is contained in the complexity class $\mathcal{NP}$, all attempts either to show that it is also contained in co-$\mathcal{NP}$ (or even that it can be solved in polynomial time) as well as all efforts in the direction of proving its $\mathcal{NP}$-completeness have failed so far. In fact, this apparent difficulty of classifying the complexity is shared by a number of isomorphism problems.

There are a variety of problems which are in a certain sense as difficult as the graph isomorphism problem, which means that they are efficiently reducible to the graph isomorphism problem and vice versa, where two concepts of reducibility are important.

A decision problem A is *Karp reducible* to another decision problem B, if there is a polynomial time algorithm which constructs from an instance I of A an instance J of B with the property that the answer for J is "yes" if and only if the answer for I is "yes." Two decision problems A and B are called *Karp equivalent* if A is Karp reducible to B and vice versa. A (decision) problem which is Karp equivalent to the graph isomorphism problem is called *graph isomorphism complete.* Often a decision problem Π is called *graph isomorphism hard* if the graph isomorphism problem is Karp reducible to Π.

Among the graph isomorphism complete problems are the restriction of the graph isomorphism problem to the class of bipartite graphs (and therefore comparability graphs), regular graphs [5,26,9], line graphs (see Harary [13]), chordal graphs [6], and self-complementary graphs [8]; even the question to decide whether a graph is self-complementary is graph isomorphism complete [8]. A recent hardness result on a problem that is not a special case of the graph isomorphism problem is due to Kutz [20]: he proved that for every $k > 0$, checking if a directed graph has a "k-th root" is graph isomorphism hard.

Other graph isomorphism complete problems occur in algebra (semi-group isomorphism [5], finitely presented algebra isomorphism [19]), as well as in topology (homeomorphism of 2-complexes [29] and homotopy equivalence [31]).

Some interesting problems related to graph isomorphism are equivalent to graph isomorphism if we use a weaker concept of reducibility: A problem A is called *Turing reducible* to a problem B if there is a polynomial time algorithm for the problem A that might use an oracle for solving B, where each call to the oracle is assumed to take only one step. Two problems A and B are *Turing equivalent* if A is Turing reducible to B and vice versa. Mathon [24] proved that a number of problems on graphs are Turing equivalent to graph isomorphism, including counting the number of *automorphisms* (i.e., isomorphisms between the graph and itself), finding a set of generators of the automorphism group, and constructing the *automorphism partition* (i.e. the orbits of the nodes under the automorphism group).

Lubiw [21] showed that the problem to decide, whether for a given graph G and two specified nodes v and w of G there is an automorphism of G not mapping v to w, is Turing equivalent to the graph isomorphism problem. In contrast to this, she showed that deciding whether a graph has a fix-point free automorphism is $\mathcal{NP}$-complete.

For further information about the graph isomorphism problem, we refer to the books by Hoffmann [14], and by Köbler, Schöning & Torán [18], as well as to the surveys by Read & Corneil [28], Babai [2], and Fortin [10].

Overview. In Sect. 2, we show that the (combinatorial) polytope isomorphism problem is graph isomorphism complete (Theorem 2). This remains true for *simple* (every vertex figure is a simplex) as well as for *simplicial* (every facet is a simplex) polytopes. The general graph isomorphism problem can be Karp-reduced to each of the three geometric polytope isomorphism problems stated above (Theorem 3). Furthermore, the graph isomorphism problem restricted to graphs of polytopes (formed by their vertices and edges) is graph isomorphism complete, even for the graphs of simple polytopes and the graphs of simplicial polytopes (Theorem 1).

In Sect. 3, we describe a polynomial time algorithm for the isomorphism problem of polytopes of bounded dimensions (Theorem 4).

2. Hardness Results for Arbitrary Dimension

The results in this section are based on the following construction that produces from any given graph $G = (V, E)$ on $n = |V|$ nodes a certain polytope $\mathcal{P}(G)$ (see Fig. 1 and Fig. 2). We denote by $\mathcal{G}(P)$ the graph of a polytope P defined by the vertices and one-dimensional faces of P.

First step. Choose an arbitrary bijection of V to the n vertices of an $(n-1)$-dimensional simplex Δ_{n-1}, thus embedding G into the graph $\mathcal{G}(\Delta_{n-1})$ of Δ_{n-1}. We call those edges of $\mathcal{G}(\Delta_{n-1})$ which are images under that embedding *black* edges, and the other ones *red* edges.

Second step. Cut off each vertex of the simplex Δ_{n-1} to obtain a polytope Γ_{n-1}. The graph $\mathcal{G}(\Gamma_{n-1})$ of Γ_{n-1} arises from $\mathcal{G}(\Delta_{n-1})$ by replacing each vertex by an $(n-1)$-clique (see Fig. 3). We call the edges of these cliques *blue* edges. Thus, $\mathcal{G}(\Gamma_{n-1})$ has black edges corresponding to the edges of G, red edges

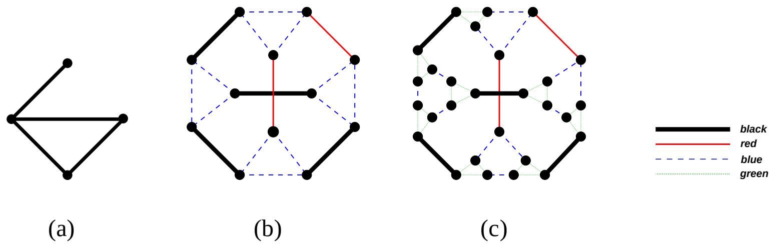


Fig. 1. Illustration of the construction of the graph $\mathcal{G}(\mathcal{P}(G))$ (depicted in (c)) from a graph G (depicted in (a)). The graph of the intermediate polytope Γ_{n-1} is shown in (b).

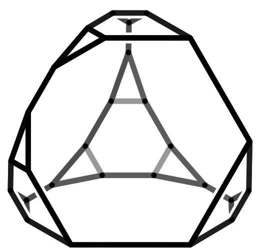


Fig. 2. The polytope $\mathcal{P}(G)$ for the graph G in Fig. 1. (Image produced with `polymake` [11] and `javaview` [27].)

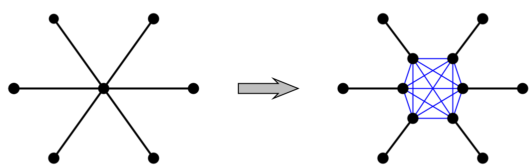


Fig. 3. Cutting off a vertex v of a simple k-polytope means to replace the node v in the graph by a k-clique.

corresponding to the edges of the complement of G, and blue edges coming from cutting off the vertices of Δ_{n-1}.

Third step. Construct $\mathcal{P}(G)$ from Γ_{n-1} by cutting off those vertices that are incident to black edges. We call the edges of the $(n-1)$-cliques that arise *green* edges.

The polytope $\mathcal{P}(G)$ is a simple $(n-1)$-polytope (see Fig. 2); its dual $\mathcal{P}(G)^\star$ is a simplicial $(n-1)$-polytope. The operation dual to cutting off a vertex of a polytope P is a *stellar subdivision* of the corresponding facet of $P^\star$, which means to place a new vertex slightly beyond this facet. Thus, $\mathcal{P}(G)^\star$ can be obtained from a simplex by iteratively placing new vertices beyond facets; it is a *stacked polytope*. In the graph of a polytope, placing a vertex beyond a facet has the effect that a new vertex is added which is connected to all vertices of that facet.

Remark 1 There are polynomial time algorithms that compute from a graph G

(i) $\mathcal{G}(\mathcal{P}(G))$ and $\mathcal{G}(\mathcal{P}(G)^\star)$,
(ii) the vertex-facet incidences of $\mathcal{P}(G)$ and $\mathcal{P}(G)^\star$,
(iii) $\mathcal{V}$-descriptions of $\mathcal{P}(G)$ and $\mathcal{P}(G)^\star$, and
(iv) $\mathcal{H}$-descriptions of $\mathcal{P}(G)$ and $\mathcal{P}(G)^\star$.

The important property of $\mathcal{P}(G)$ is that it encodes the entire structure of G.

Proposition 1 *For two graphs G and H on at least three nodes the following five statements are equivalent.*

(i) G is isomorphic to H.
(ii) $\mathcal{G}(\mathcal{P}(G))$ is isomorphic to $\mathcal{G}(\mathcal{P}(H))$.
(iii) $\mathcal{G}(\mathcal{P}(G)^\star)$ is isomorphic to $\mathcal{G}(\mathcal{P}(H)^\star)$.
(iv) $\mathcal{P}(G)$ is isomorphic to $\mathcal{P}(H)$.
(v) $\mathcal{P}(G)^\star$ is isomorphic to $\mathcal{P}(H)^\star$.

Proof. We start by proving the equivalence of (i) and (ii).

Any isomorphism between two graphs G and H induces a color preserving isomorphism between the two complete graphs constructed from G and H in the first step. Of course, such a color preserving isomorphism induces a color preserving isomorphism of the graphs of the polytopes constructed in the second step, which finally gives rise to an isomorphism of the graphs $\mathcal{G}(\mathcal{P}(G))$ and $\mathcal{G}(\mathcal{P}(H))$ of the two polytopes constructed in the third step.

In order to prove the converse direction, let G and H be two graphs on n and n' nodes ($n, n' \geq 3$), respectively, and let φ be an isomorphism between $\mathcal{G}(\mathcal{P}(G))$ and $\mathcal{G}(\mathcal{P}(H))$. Since $\mathcal{G}(\mathcal{P}(G))$ is $(n-1)$-regular and $\mathcal{G}(\mathcal{P}(H))$ is $(n'-1)$-regular, we have $n = n'$. If $n = 3$, then both $\mathcal{G}(\mathcal{P}(G))$ and $\mathcal{G}(\mathcal{P}(H))$ are cycles of length ℓ. Since in this case, the number of edges of G as well as of H must be $(\ell - 6)/2 \in \{0, 1, 2, 3\}$, G and H are isomorphic. Thus, we may assume $n \geq 4$.

We consider $\mathcal{G}(\mathcal{P}(G))$ and $\mathcal{G}(\mathcal{P}(H))$ colored as defined in the description of the construction. In both graphs, all $(n-1)$-cliques are node-disjoint. Each of these cliques either consists of green or of blue edges (blue cliques might arise from isolated nodes). Consider the graphs that arise from $\mathcal{G}(\mathcal{P}(G))$ and $\mathcal{G}(\mathcal{P}(H))$ by shrinking all $(n-1)$-cliques. Those nodes that come from shrinking green cliques are contained in (maximal) $(n-1)$-cliques in the shrunken graphs, while those coming from blue cliques are not (notice that for graphs without edges this statement indeed only holds for *maximal* $(n-1)$-cliques). This shows that φ preserves the colors of $(n-1)$-cliques.

Let G' and H' be the graphs that are obtained from shrinking the *green* cliques in $\mathcal{G}(\mathcal{P}(G))$ and $\mathcal{G}(\mathcal{P}(H))$, respectively. Since φ maps green cliques to green cliques, it induces an isomorphism ψ between G' and H'. Since the shrinking operations do not generate multiple edges, the graphs G' and H' inherit colorings of their edges from $\mathcal{G}(\mathcal{P}(G))$ and $\mathcal{G}(\mathcal{P}(H))$, respectively. Because an edge of $\mathcal{G}(\mathcal{P}(G))$ or $\mathcal{G}(\mathcal{P}(H))$ is red if and only if it is not adjacent to a green edge, the isomorphism ψ preserves red edges.

In the graphs G' and H' the only $(n-1)$-cliques are the ones formed by the blue edges. Again, these cliques are pairwise node-disjoint. Thus the isomorphism ψ between G' and H' induces a color preserving isomorphism between the (complete) graphs obtained by shrinking all $(n-1)$-cliques in G' and H' (which, again, does not produce multiple edges). This, finally, yields an isomorphism between G and H.

The equivalence of (ii) and (iv) follows from a theorem of Blind and Mani [4] (see also Kalai's beautiful proof [17]) stating that two simple polytopes are isomorphic if and only if their graphs are isomorphic. For the special polytopes arising from our construction, the equivalence can, however, be alternatively deduced similarly to the proof of "(ii) $\Rightarrow$ (i)."

Statements (iv) and (v) obviously are equivalent.

Unlike the situation for simple polytopes, it is, in general, not true that two (simplicial) polytopes are isomorphic if and only their graphs are isomorphic. Nevertheless, for stacked polytopes like $\mathcal{P}(G)^{\star}$ and $\mathcal{P}(H)^{\star}$ it is true (this follows, e.g., from the fact that one can reconstruct the vertex-facet incidences of a stacked d-polytope from its graph by iteratively removing vertices of degree d). Thus, finally the equivalence of (iii) and (v) is established. □

The equivalences "(i) $\Leftrightarrow$ (ii)" and "(i) $\Leftrightarrow$ (iii)" in Proposition 1 together with part (i) of Remark 1 immediately imply that the restriction to graphs of simple or of simplicial polytopes does not make the graph isomorphism problem easier.

Theorem 1 *The graph isomorphism problem restricted to graphs of polytopes is graph isomorphism complete, even if one restricts the problem further to the class of graphs of simple or to the class of graphs of simplicial polytopes.*

In particular, Theorem 1 for simple polytopes implies that the restricition of the graph isomorphism problem to regular graphs is graph isomorphism complete. As mentioned in the introduction, this is well-known. However, the different reductions due to Booth [5], Miller [26], and Corneil & Kirkpatrick [9] do not produce polytopal graphs.

The equivalences "(i) $\Leftrightarrow$ (iv)" and "(i) $\Leftrightarrow$ (v)" in Proposition 1 together with part (ii) of Remark 1 also imply the main result of this section.

Theorem 2 *The polytope isomorphism problem is graph isomorphism complete, even if one restricts the problem further to the class of simple or to the class of simplicial polytopes.*

In fact, since the duals of the polytopes $\mathcal{P}(G)$ are stacked polytopes, Theorems 1 and 2 even hold for the very restricted class of stacked polytopes.

Parts (iii) and (iv) of Remark 1 show that the polytope isomorphism problem remains graph isomorphism complete if additionally $\mathcal{V}$- and $\mathcal{H}$-descriptions of the polytopes are provided as input data. Even more: if two graphs G and H are isomorphic, then the polytopes $\mathcal{P}(G)$ and $\mathcal{P}(H)$ are affinely isomorphic (here, of

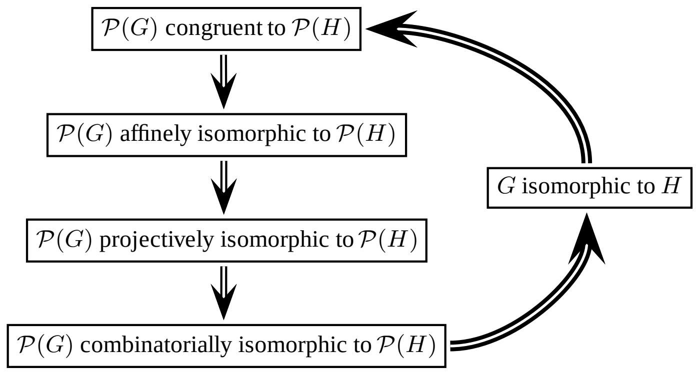


Fig. 4. Illustration of the proof of Theorem 3. The two long arrows show implications following from Proposition 1 and the remarks above, while the three other arrows represent implications that hold in general.

course, all cutting operations have to be performed "in the same way"). If we start our constructions with the regular $(n-1)$-dimensional simplex

$$\text{conv}\{(1,0,\ldots,0),(0,1,0,\ldots,0),\ldots,(0,\ldots,0,1)\} \subset \mathbb{R}^n$$

embedded into $\mathbb{R}^n$, then we can furthermore achieve that $\mathcal{P}(G)$ and $\mathcal{P}(H)$ are congruent if G and H are isomorphic. Consequently, Proposition 1 ("(iv) $\Rightarrow$ (i)") shows that the implications depicted by the arcs in Figure 4 hold.

Thus, all statements in that figure are pairwise equivalent, and therefore, we have proved the following result on the geometric polytope isomorphism problems.

Theorem 3 *The graph isomorphism problem is Karp reducible to the polytope congruence problem as well as to the affine and to the projective polytope isomorphism problem (with respect to both $\mathcal{V}$- and $\mathcal{H}$-descriptions). This remains true for the restrictions to simple or to simplicial polytopes.*

It has been proved by Akutsu [1] that the graph isomorphism problem is Karp reducible to the congruence problem for arbitray point sets which are not necessarily in convex position. By an obvious modification his construction can be changed to produce point sets in convex position, thus giving an alternative proof of the graph isomorphism hardness of polytope congruence. The resulting reduction also shows the graph isomorphism hardness of the affine and of the projective polytope isomorphism problem. However, this construction does neither yield simple nor simplicial polytopes. Moreover, it is not clear that combinatorial isomorphism between two constructed polytopes implies that the graphs started from are isomorphic as well. Hence, Akutsu's construction does not provide an alternative proof of Theorem 2.

Akutsu [1] also gave a Karp reduction of the congruence problem to the (labeled) graph isomorphism problem, thus showing that the former one is graph isomorphism complete. It is, however, unknown if the affine or the projective polytope isomorphism problem can be reduced to the graph isomorphism problem as well.

3. Polynomiality Results for Bounded Dimension

For polytopes of dimension one or two both the graph isomorphism problem as well as the polytope isomorphism problem can obviously be solved in polynomial time, even in linear time. It is well-known that two three-dimensional polytopes are combinatorially isomorphic if and only if their graphs are isomorphic (this follows from the theorem of Whitney [30] on the uniqueness of the plane embedding of a planar three-connected graph). Since the graph isomorphism problem for planar graphs can be solved in linear time by an algorithm due to Hopcroft and Wong [15], both the graph and the polytope isomorphism problem for three-dimensional polytopes can thus be solved in linear time.

The main result of this section is that, for every bounded dimension, the polytope isomorphism problem can be solved in polynomial time. First, we consider the polytope isomorphism problem restricted to simple polytopes.

If two polytopes are combinatorially isomorphic, then they have the same dimension as well as the same number of vertices and the same number of facets. It is trivial to determine these three parameters from the vertex-facet incidences. Therefore, we will assume in the formulations of our algorithms for the polytope isomorphism problem that each of the three parameters has the same value for both input polytopes.

In the following, we denote the (abstract) sets of vertices and edges of a polytope P by $\mathrm{V}(P)$ and $\mathrm{E}(P)$, respectively.

Proposition 2 *The polytope isomorphism problem for simple d-polytopes can be solved in $\mathcal{O}(d! \cdot d^2 \cdot n^2)$ time, where n is the number of vertices. In particular, there is an algorithm for the problem whose running time is $\mathcal{O}(n^2)$ for bounded d.*

Proof. The key observation for the proof is that every isomorphism of two simple polytopes is determined by its restriction to an arbitrary vertex and its neighborhood.

Throughout the following, assume $d > 2$. Let P be a simple d-polytope. We denote by $\mathrm{N}(x)$ the set of neighbors of the node x in $\mathcal{G}(P)$ and define $\overline{\mathrm{N}}(x) := \mathrm{N}(x) \cup \{x\}$. The 2-skeleton of P induces, for each edge $\{v, w\} \in \mathrm{E}(P)$ of P, a bijection $\Psi_{v,w} \colon \overline{\mathrm{N}}(v) \longrightarrow \overline{\mathrm{N}}(w)$ with $\Psi_{v,w}(v) = v$, $\Psi_{v,w}(w) = w$, and $\Psi_{v,w}(u)$ being the other (than v) neighbor of w in the 2-face spanned by v, w, and u.

Suppose $\pi \colon \mathrm{V}(P) \longrightarrow \mathrm{V}(Q)$ is an isomorphism of the graphs $\mathcal{G}(P)$ and $\mathcal{G}(Q)$ of two simple d-polytopes P and Q, respectively. For each $v \in \mathrm{V}(P)$ we denote by $\pi_v \colon \overline{\mathrm{N}}(v) \longrightarrow \mathrm{V}(Q)$ the restriction of π to $\overline{\mathrm{N}}(v)$. For every node $v \in \mathrm{V}(P)$, we have

$$\pi_v(\mathrm{N}(v)) = \mathrm{N}(\pi_v(v)) \quad , \tag{1}$$

and, for every edge $\{v,w\} \in \mathrm{E}(P)$,

$$\pi_w = \Psi_{\pi_v(v),\pi_v(w)} \circ \pi_v \circ \Psi_{w,v} \tag{2}$$

(where the composition $\Psi_{\pi_v(v),\pi_v(w)} \circ \pi_v$ is well-defined due to (1)).

Conversely, for two simple d-polytopes P and Q with n vertices consider any set of maps $\pi_v\colon \overline{\mathrm{N}}(v) \longrightarrow \mathrm{V}(Q)$ ($v \in \mathrm{V}(P)$). If the maps are consistent, i.e.,

$$\pi_v(u) = \pi_w(u) \qquad (v,w,u \in \mathrm{V}(P), u \in \overline{\mathrm{N}}(v) \cap \overline{\mathrm{N}}(w)) \ , \tag{3}$$

then there is a unique map $\pi\colon\ \mathrm{V}(P) \longrightarrow \mathrm{V}(Q)$ such that π_v is the restriction of π to $\overline{\mathrm{N}}(v)$ for all $v \in \mathrm{V}(P)$. We claim that, if π satisfies (1) for all $v \in \mathrm{V}(P)$, then π is an isomorphism of $\mathcal{G}(P)$ and $\mathcal{G}(Q)$ (and thus a combinatorial isomorphism between P and Q). To see this, it suffices to show that π is surjective, since both $\mathcal{G}(P)$ and $\mathcal{G}(Q)$ have the same number n of nodes. Suppose that π is not surjective. Since $\mathcal{G}(Q)$ is connected, there is a node y of $\mathcal{G}(Q)$ which is not contained in the image of π and which has a neighbor x that is the image $\pi(v)$ of some node v of $\mathcal{G}(P)$. However, this contradicts (1).

Thus, we may check two simple d-polytopes P and Q both with n vertices, given by their vertex-facet incidences, for combinatorial isomorphism in the following way. First, we compute the graphs $\mathcal{G}(P)$ and $\mathcal{G}(Q)$, as well as the bijections $\Psi_{v,w}$ ($\{v,w\} \in \mathrm{E}(P)$) and $\Psi_{x,y}$ ($\{x,y\} \in \mathrm{E}(Q)$). Furthermore, a node $v_0 \in \mathrm{V}(P)$ is fixed together with a spanning tree T_0 of $\mathcal{G}(P)$, rooted at v_0.

Then, for each node $x \in \mathrm{V}(Q)$ and for each bijection $\pi_{v_0}\colon \overline{\mathrm{N}}(v_0) \longrightarrow \overline{\mathrm{N}}(x)$ with $\pi_{v_0}(v_0) = x$, we perform the following steps:

1. Compute π_w ($w \in \mathrm{V}(P)$) "along T_0" by means of (2); if, for some w, condition (1) is not satisfied, then continue with the next bijection π_{v_0}.
2. If (3) is not satisfied, then continue with the next bijection π_{v_0}.
3. Construct the map π (as above); STOP.

Note that, when π_w is computed in Step 1 from the parent v of w in T_0, condition (1) is satisfied for v (thus, the composition in (2) is well-defined). It follows from the discussion above that the two polytopes are isomorphic if and only if the algorithm stops in Step 3; in this case, the constructed π is an isomorphism between P and Q.

Computing the graphs $\mathcal{G}(P)$ and $\mathcal{G}(Q)$, as well as the bijections $\Psi_{v,w}$ ($\{v,w\} \in \mathrm{E}(P)$) and $\Psi_{x,y}$ ($\{x,y\} \in \mathrm{E}(Q)$), can be performed in $\mathcal{O}(d \cdot n^2)$ time, if we perform the following preprocessing in advance. From the vertex-facet incidences compute, for each vertex, a sorted list of indices of the facets containing this vertex. This can be performed in $\mathcal{O}(n^2)$ steps (note that simple polytopes never have more facets than vertices). Compute a similar incidence list for each facet.

Moreover, using this data structure, none of the four steps in the for-loop needs more than $\mathcal{O}(d^2 \cdot n)$ time (the critical part being Step 2). Since the body of the for-loop is not executed more than $n \cdot d!$ times, this yields an $\mathcal{O}(d! \cdot d^2 \cdot n^2)$ time algorithm. □

Luks [22] gave a polynomial time algorithm for the graph isomorphism problem on graphs of bounded maximal degree. Since the graph of a simple d-polytope is d-regular, his algorithm runs in polynomial time on graphs of simple polytopes of bounded dimension.

Proposition 3 *The isomorphism problem for graphs of simple polytopes of bounded dimension can be solved in polynomial time.*

As two simple polytopes are isomorphic if and only if their graphs are isomorphic (cf. the proof of Proposition 1) and since it is easy to compute efficiently the graph of a polytope from its vertex-facet incidences, Proposition 3 implies the polynomiality statement of Proposition 2. The algorithm described in the proof of Proposition 2, however, is both much simpler to understand/implement, and much faster than Luks' method. The running time of the original version of Luks' isomorphism test for graphs of bounded maximal degree d is $n^{\mathcal{O}(d^3)}$, where n is the number of nodes. According to Luks, this could be improved to $n^{O(d/\log d)}$ [23], which, in general, is much larger than $\mathcal{O}(n^2)$.

Two polytopes are isomorphic if and only if their dual polytopes are isomorphic. Since the transpose of a vertex-facet incidence matrix of a polytope is a vertex-facet incidence matrix of the dual polytope, Proposition 2 implies its own analogue for simplicial polytopes.

Proposition 4 *The polytope isomorphism problem for simplicial d-polytopes can be solved in $\mathcal{O}(d! \cdot d^2 \cdot m^2)$ time, where m is the number of facets. In particular, there is an algorithm for the problem whose running time is $\mathcal{O}(m^2)$ for bounded d.*

While simple and simplicial polytopes play symmetric roles with respect to the polytope isomorphism problem, they may play different roles with respect to the graph isomorphism problem. In particular, it is unknown whether Proposition 3 is also true for the graphs of simplicial polytopes of bounded dimensions (see Sect. 4).

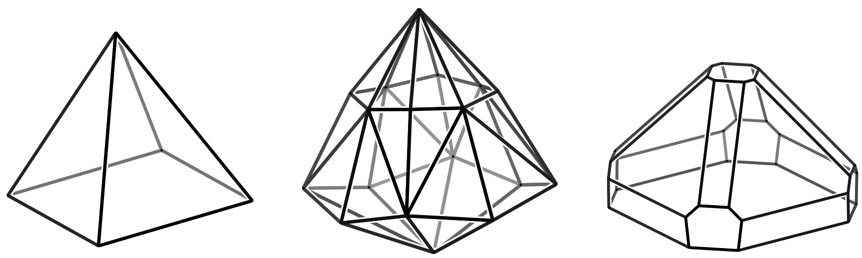
The main result of this section is that one can extend Propositions 2 and 4 to arbitrary polytopes (of bounded dimension).

Theorem 4 *The polytope isomorphism problem can be solved in*

$$\mathcal{O}(d^2 \cdot 2^{d^2} \cdot \min\{n,m\}^{d^2/2}) \quad \textit{time},$$

where d, n and m are the dimension, the number of vertices, and the number of facets, respectively. In particular, there is an algorithm for the problem whose running time is $\mathcal{O}(\min\{n,m\}^{d^2/2})$ for bounded d.

Proof. The core of the proof is a result of Bayer [3, see Theorem 3 and its proof] implying that every combinatorial isomorphism between the barycentric subdivisions of two polytopes P and Q induces a combinatorial isomorphism between P and Q or its dual polytope $Q^\star$, where the *barycentric subdivision* of a d-polytope R may be defined as any d-polytope $\beta(R)$, whose vertices correspond to the non-trivial faces ($\neq \varnothing, R$) of R, and whose facets correspond to the maximal chains in the face lattice of R. Such a polytope $\beta(R)$ can be constructed from R by performing stellar subdivisions of all k-faces for $k = d-1, \ldots, 1$ (see Fig. 5).



(a) R (b) $\beta(R)$ (c) $\beta(R)^\star$

Fig. 5. The barycentric subdivision $\beta(R)$ of a 3-dimensional pyramid R and the dual polytope $\beta(R)^\star$.

The dual $\beta(R)^\star$ is a simple d-polytope whose vertices correspond to the maximal chains in the face lattice of R, and whose facets correspond to the (non-empty) faces of R. In the graph of $\beta(R)^\star$, two nodes are adjacent if and only if the corresponding two maximal chains of R arise from each other by exchanging the two i-dimensional faces (for some unique $0 \leq i \leq d-1$). This defines a labeling of the edges of $\mathcal{G}(\beta(R)^\star)$ by $\{0,1,\ldots,d-1\}$, where for each node the d incident edges receive pairwise different labels. Thus, Bayer's result implies that two polytopes P and Q are combinatorially isomorphic if and only if there is a combinatorial isomorphism between $\beta(P)^\star$ and $\beta(Q)^\star$ that induces a label preserving isomorphism of $\mathcal{G}(\beta(P)^\star)$ and $\mathcal{G}(\beta(Q)^\star)$.

Let P and Q be two d-polytopes, given by their vertex-facet incidences, which are to be checked for combinatorial isomorphism. We first compute the Hasse diagram of the face lattice of each of the two polytopes (with nodes labeled by the dimensions of the respective faces). From the Hasse diagrams we then enumerate the maximal chains of the face lattices of P and Q, the vertex-facet incidences of $\beta(P)^\star$ and $\beta(Q)^\star$, as well as the graphs $\mathcal{G}(\beta(P)^\star)$ and $\mathcal{G}(\beta(Q)^\star)$ together with their edge labelings.

Once these data are available, we can check $\beta(P)^\star$ and $\beta(Q)^\star$ for combinatorial isomorphism as in the proof of Proposition 2. However, since we only allow isomorphisms that respect the edge labeling of $\mathcal{G}(\beta(P)^\star)$ and $\mathcal{G}(\beta(Q)^\star)$, for each potential image x of the fixed vertex v_0 of $\beta(P)^\star$ there is only one possible bijection between $\overline{\mathrm{N}}(v_0) \longrightarrow \overline{\mathrm{N}}(x)$ (with the notation adapted from the proof of Proposition 2). Consequently, the running time of checking $\beta(P)^\star$ and $\beta(Q)^\star$ for (label preserving) combinatorial isomorphism can be estimated by $\mathcal{O}(d^2 \cdot \zeta^2)$, where ζ is the number of maximal chains in the face lattice of P or Q. We may assume that ζ is the same number for both P and Q, since otherwise we know already that P is not isomorphic to Q; the same holds for the numbers φ, α, n, and m arising below.

Computing the Hasse diagrams of P and Q can be performed in time $\mathcal{O}(\varphi \cdot \alpha \cdot \min\{n,m\})$ (see [16]), where φ, α, n, and m are the number of faces, the number of vertex-facet incidences, the number of vertices, and the number of facets of P or Q, respectively. Enumerating all maximal chains in the face lattices of P and Q, can be done in time proportional to $d \cdot \zeta$. Computing the vertex-facet incidences of $\beta(P)^\star$ and $\beta(Q)^\star$ takes no more than $\mathcal{O}(\varphi \cdot \zeta)$ steps. The edge labeled graphs $\mathcal{G}(\beta(P)^\star)$ and $\mathcal{G}(\beta(Q)^\star)$ as well as the maps $\Psi_{\star,\star}$ can be obtained (see the proof of Proposition 2) in time $\mathcal{O}(d \cdot \zeta^2)$ (note that $\Psi_{\star,\star}$ can easily be obtained from the edge labeling).

By the upper bound theorem for convex polytopes (see, e.g., [32, Thm. 8.23]) applied to the barycentric subdivisions, we have $\zeta \in \mathcal{O}(\varphi^{d/2})$, where, again by the upper bound theorem, we can estimate $\varphi \leq 2^d \cdot \min\{n, m\}^{d/2}$. In total, the running time of the sketched algorithm thus can be bounded by $\mathcal{O}(d^2 \cdot \zeta^2) = \mathcal{O}(d^2 \cdot 2^{d^2} \cdot \min\{n, m\}^{d^2/2})$. □

The estimate for the running time in this proof uses the upper bound theorem twice in a nested way. Thus, in practice the running time will usually be much smaller than the estimate of the theorem.

All three geometric polytope isomorphism problems are polynomially solvable in bounded dimensions: If the vertices $\operatorname{vert}(P) \subset \mathbb{R}^d$ and $\operatorname{vert}(Q) \subset \mathbb{R}^d$ of two polytopes P and Q are given, one may first choose a maximal affinely independent set $S \subset \operatorname{vert}(P)$. Then, for each maximal affinely independent set $T \subset \operatorname{vert}(Q)$ and for each bijection $\pi\colon S \longrightarrow T$ one checks whether the affine map $\operatorname{aff}(P) \longrightarrow \operatorname{aff}(Q)$ induced by π satisfies $\pi(\operatorname{vert}(P)) = \operatorname{vert}(Q)$. Similar algorithms can be constructed for checking congruence or projective isomorphism. If the polytopes are specified by $\mathcal{H}$-descriptions one may first compute $\mathcal{V}$-representations (in polynomial time, since the dimension is bounded).

There are more elaborate algorithms for checking two sets of n points in $\mathbb{R}^d$ for congruence. For instance, Brass & Knauer [7] describe a deterministic $\mathcal{O}(n^{\lceil d/3 \rceil} \cdot \log n)$ algorithm.

4. Conclusions

The main results of this paper are on the one hand the graph isomorphism completeness of the general (combinatorial) polytope isomorphism problem and, on the other hand, the fact that this problem can be solved in polynomial time if the dimensions of the polytopes are bounded by a constant (see Table 1 for an overview of the complexity results and Fig. 6 for a sketch of the complexity theoretic landscape considered in this paper).

Our hardness result for arbitrary dimensions leaves open the question for the complexity status of the problem to decide whether two polytopes given by their entire face lattices (rather than by their vertex-facet incidences only) are isomorphic (*face lattice isomorphism problem*). The algorithm described in the proof of Theorem 4 does not solve the face lattice isomorphism problem in polynomial time, since the number of maximal chains in the face lattice of a polytope is not bounded polynomially in the size of the face lattice.

The polynomiality result on the polytope isomorphism problem in bounded dimensions (Theorem 4) theoretically supports the empirical evidence that check-

Table 1. Summary of the complexity results (GI means graph isomorphism).

		polytopal graph isomorphism	polytope isomorphism	
			incidences	face lattice
bounded dimension	simplicial	*open*	polynomial	polynomial
	simple	polynomial	polynomial	polynomial
	arbitrary	*open*	polynomial	polynomial
arbitrary dimension	simplicial	GI complete	GI complete	*open*
	simple	GI complete	GI complete	*open*
	arbitrary	GI complete	GI complete	*open*

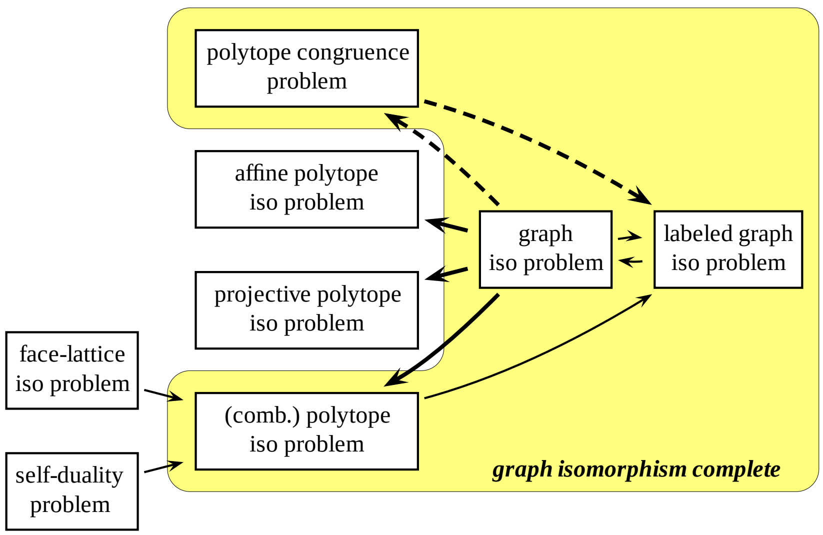


Fig. 6. Illustration of several decision problems treated in the paper; each arrow means that there is a Karp-reduction from the problem at its tail to the problem at its head. The solid bold arrows indicate new reductions and the two dashed ones emerge from Akutsu [1].

ing whether two polytopes of moderate dimensions are isomorphic is not too hard. This evidence stems from applying McKay's `nauty` to the vertex-facet incidences as mentioned in the introduction. It may be that one can turn our algorithm into a computer code that becomes compatible with `nauty` for checking combinatorial polytope isomorphism.

The remaining two *open* entries in Table 1 concern the complexity of the graph isomorphism problem restricted to graphs of arbitrary (or simplicial) polytopes of bounded dimensions. A polynomial time algorithm for this problem would perhaps not be as interesting as the potential result that the problem is graph isomorphism complete, because the latter result would show that the class of graphs of polytopes of any fixed dimension is in a sense "structurally as rich" as the class of all graphs.

If the face lattice isomorphism problem turned out to be graph isomorphism complete, then this would immediately imply that the face lattice isomorphism problem is also "(finite) poset isomorphism complete." Similarly to the case of the graphs of polytopes of fixed dimensions, one might interpret such a result as an evidence that the class of face lattices form a "structurally rich" subclass of the class of all (finite) posets.

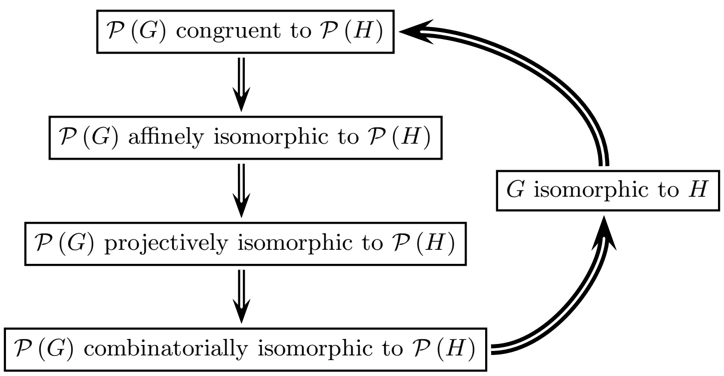
An interesting special case of the combinatorial polytope isomorphism problem is the *polytope self-duality problem*, which asks to decide whether a polytope, given by its vertex-facet incidences, is combinatorially isomorphic to its dual. Clearly, by Theorem 4 this problem can be solved in polynomial time for bounded dimensions. However, its general complexity status remains open.

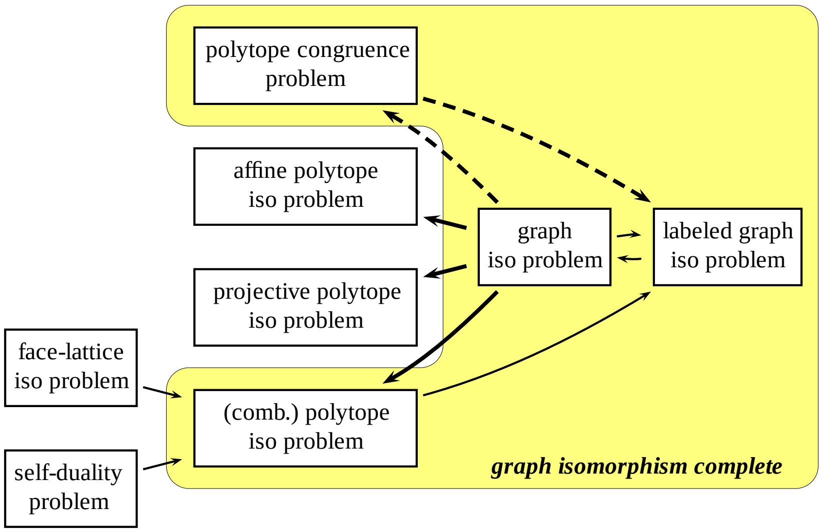
Acknowledgements. We are indebted to Günter M. Ziegler for careful reading an earlier version of the paper and for stimulating discussions, in particular, for bringing Bayer's work [3] to our attention. We thank Günter Rote for inspiring comments, Christian Knauer for pointing to Akutsu's results [1], and Christoph Eyrich for valuable LaTeX advice.

References

1. Akutsu, T.: On determining the congruence of point sets in d dimensions. Comput. Geom. **9**, 247–256 (1998)
2. Babai, L.: Automorphism groups, isomorphism, reconstruction. In: R.L. Graham et al. (eds.): Handbook of Cominatorics (pp. 1447–1540), Amsterdam, North-Holland: Elsevier 1995
3. Bayer, M.: Barycentric subdivisions. Pac. J. Math. **135**, 1–17 (1988)
4. Blind, R. and Mani-Levitska, P.: On puzzles and polytope isomorphisms. Aequationes Math. **34**, 287–297 (1987)
5. Booth, K.: Isomorphism testing for graphs, semigroups, and finite automata are polynomially equivalent problems. SIAM J. Comput. **7**, 273–279 (1978)
6. Booth, K. and Lueker, G. S.: Linear algorithms to recognize interval graphs and test for the consecutive ones property. Proc. 7th Ann. ACM Symp. Theory Comput., 255–265 (1975)
7. Brass, P. and Knauer, C.: Testing the congruence of d-dimensional point sets. J. Comput. Geom. Appl. **12**, 115–124 (2002)
8. Colbourn, M.J. and Colbourn, J.: Graph isomorphism and self-complementary graphs. SIGACT News **10**, 25–30 (1978)
9. Corneil, D. and Kirkpatrick, D.: A theoretical analysis of various heuristics for the graph isomorphism problem. SIAM J. Comput. **9**, 281–297 (1980)
10. Fortin, S.: The graph isomorphism problem, tech. report TR 96-20, Dept. of Computer Science, University of Alberta, Edmonton, Alberta, Canada, 1996 `ftp://ftp.cs.ualberta.ca/pub/TechReports/1996/TR96- 20/ TR96-20.ps.gz`
11. Gawrilow, E. and Joswig, M.: polymake: A framework for analyzing convex polytopes. In: G. Kalai and G.M. Ziegler (eds.): Polytopes — Combinatorics and Computation (DMV-Seminars, pp. 43–74) Basel: Birkhäuser-Verlag Basel 2000, see also `http://www.math.tu-berlin.de/diskregeom/polymake`
12. Grünbaum, B.: Convex Polytopes. New York: Interscience Publishers John Wiley & Sons 1967

13. Harary, F.: Graph Theory. Reading, MA: Addison-Wesley 1969
14. Hoffmann, C.: Group-theoretic Algorithms and Graph Isomorphism (Lecture Notes in Computer Science 136) Berlin-Heidelberg-New York: Springer 1982
15. Hopcroft, J. and Wong, J.: A linear time algorithm for isomorphism of planar graphs. Proc. 6th Ann. ACM Symp. Theory Comput., 172–184 (1974)
16. Kaibel, V. and Pfetsch, M.E.: Computing the face lattice of a polytope from its vertex-facet incidences. Comput. Geom. (to appear)
17. Kalai, G.: A simple way to tell a simple polytope from its graph. J. Comb. Theory Ser. A **49**, 381–383 (1988)
18. Köbler, J., Schöning, U., and Torán, J.: The Graph Isomorphism Problem: Its Structural Complexity (Progress in Theoretical Computer Science) Boston, MA: Birkhäuser 1993
19. Kozen, D.: On the complexity of finitely presented algebras. Proc. 9th Ann. ACM Symp. Theory Comput., 164–177 (1977)
20. Kutz, M.: Computing roots of directed graphs is graph isomorphism hard, tech. report, FU Berlin, 2002. `arXiv:math.CO/0207020`
21. Lubiw, A.: Some $\mathcal{NP}$-complete problems similar to graph isomorphism. SIAM J. Comput. **10**, 11–21 (1981)
22. Luks, E.: Isomorphism of graphs of bounded valence can be tested in polynomial time. J. Comput. System Sci. **25**, 42–65 (1982)
23. Luks, E.: Personal communication, May 2001
24. Mathon, R.: A note on the graph isomorphism counting problem. Inf. Process. Lett. **8**, 131–132 (1979)
25. McKay, B.: Practical graph isomorphism. Numerical mathematics and computing, Proc. 10th Manitoba Conf., Winnipeg/Manitoba 1980, Congr. Numerantium 30, 45–87 (1981), see also `http://cs.anu.edu.au/people/bdm/nauty`
26. Miller, G.L.: Graph isomorphism, general remarks. J. Comput. System Sci. **18**, 128–142 (1979)
27. Polthier, K., Khadem, S., Peuß, E. and Reitebuch, U.: Publication of interactive visualizations with `JavaView`. To appear in: J. Borwein et al. (eds.): Multimedia Tools for Communicating Mathematics, Springer 2001,
see also `http://www.javaview.de/`
28. Read, R. and Corneil, D.: The graph isomorphism disease. J. Graph Theory **1**, 339–363 (1977)
29. Shawe-Taylor, J. and Pisanski, T.: Homeomorphism of 2-complexes is graph isomorphism complete. SIAM J. Comput. **23**, 120–132 (1994)
30. Whitney, H.: Congruent graphs and the connectivity of graphs. Am. J. Math. **54**, 150–168 (1932)
31. Zalcstein, Y. and Franklin, S.P.: Testing homotopy equivalence is isomorphism complete. Discrete Appl. Math. **13**, 101–104 (1986)
32. Ziegler, G.M.: Lectures on Polytopes (Graduate Texts in Mathematics 152) New York: Springer 1995, revised edition 1998


$\mathcal{P}(G)$ congruent to $\mathcal{P}(H)$
$\mathcal{P}(G)$ affinely isomorphic to $\mathcal{P}(H)$
$\mathcal{P}(G)$ projectively isomorphic to $\mathcal{P}(H)$
$\mathcal{P}(G)$ combinatorially isomorphic to $\mathcal{P}(H)$
G isomorphic to H



polytope congruence problem
affine polytope iso problem
graph iso problem
labeled graph iso problem
projective polytope iso problem
face-lattice iso problem
(comb.) polytope iso problem
self-duality problem
graph isomorphism complete