Small k-pyramids and the complexity of determining k

BORIS SCHAUERTE and CAROL T. ZAMFIRESCU

Abstract. Motivated by the computational complexity of determining whether a graph is hamiltonian, we study under algorithmic aspects a class of polyhedra called k-pyramids, introduced in [Zamfirescu and Zamfirescu, *Math. Nachr.* **284** (2011) 1739–1747], and discuss related applications. We prove that determining whether a given graph is the 1-skeleton of a k-pyramid, and if so whether it is belted or not, can be done in polynomial time for $k \leq 3$. The impact on hamiltonicity follows from the traceability of all 2-pyramids and non-belted 3-pyramids, and from the hamiltonicity of all non-belted 2-pyramids. The algorithm can also be used to determine the outcome for larger values of k, but the complexity increases exponentially with k. Lastly, we present applications of the algorithm, and improve the known bounds for the minimal cardinality of systems of bases called foundations in graph families with interesting properties concerning traceability and hamiltonicity.

Key Words. Pyramid, prism, Halin graph, hamiltonian.

MSC 2010. 05C45, 05C85.

1 Introduction

Determining whether a given graph is hamiltonian is a classical NP-complete problem [Karp, 1972]. Based on this, [Garey et al., 1976] showed that determining traceability is an NP-complete problem, too. Even restricted to planar, cubic, 3-connected graphs, determining hamiltonicity remains NP-complete [Garey et al., 1976]. Thus, it is also NP-complete for the class of polyhedral graphs. In this contribution, we focus on k-pyramids, a class of polyhedra which generalizes those having Halin graphs as 1-skeleta and includes pyramids and prisms. Other generalizations on Halin graphs and investigations of their hamiltonian properties have already been made by Skowrońska [1983], Skowrońska and Sysło [1987], Skupień [1990], and Malik et al. [2009].

All graphs in this paper are finite, undirected, connected and contain neither loops nor edges. Such a graph is called *polyhedral* if it is planar and 3-connected. (Recall that there is a bijection between polyhedral graphs and the 1-skeleta of polytopes by Steinitz's famous Theorem [Steinitz, 1922].) In order to study a polyhedral graph G, we will need the concepts of dual graph and independent dominating set. On one hand, constructing the dual graph is, algorithmically speaking, easily dealt with. On the other hand, it was shown by Garey and Johnson [1979] that the problem of finding an independent dominating set of minimal cardinality (MIDS) is NP-complete. It remains NP-complete if restricted to line graphs [Yannakakis and Gavril, 1980], bipartite graphs [Corneil and Perl, 1984; Irving, 1991] and dually chordal graphs [Brandstädt et al., 1998]. Polynomial-time algorithms exist for many families of graphs, e.g. chordal graphs [Farber, 1982], interval and circular-arc graphs [Chang, 1998], cocomparability graphs [Kratsch and Stewart, 1993], asteroidal triple-free graphs [Broersma et al., 1997], and series-parallel graphs [Pfaff et al., 1984; Grinstead and Slater, 1994]. For more details, see the excellent article by Manlove [1999]. In the same paper it was shown that MIDS, even restricted to cubic planar graphs, still is NP-complete.

There exist several results concerning exact exponential time algorithms for MIDS. For general graphs with n vertices we have $O(1.3575^n)$, given in [Gaspers and Liedloff, 2007], and $O^*(3^{n/2})$ [Liu and Song, 2006]. In the same paper they prove that for graphs with degree bounded by 3, we have $O^*(2^{0.465n})$.

Motivated by the computational complexity of determining hamiltonicity and traceability, and problem 3 raised in [Zamfirescu and Zamfirescu, 2011], we present two algorithms with $O(n^4)$ time complexity that, given a connected finite graph without loops and multiple edges, (i) determine the minimal value k for which the graph is the 1-skeleton of a k-pyramid, and (ii) whether the graph is belted or not. In the remainder of this paper, we will address the following subsequent problems for a given polyhedral input graph G:

(P1) Is there a natural number k such that G is isomorphic to a k-pyramid?
(P2) Compute k^*, the minimal k for which (P1) holds.
(P3) Let (P2) be satisfied, and $k^* \in \{2,3\}$. Is G non-belted?

2 Definitions

A polytope P in $\mathbb{R}^3$ is said to be *hamiltonian* (*traceable*), if its 1-skeleton – which is a polyhedral graph – has a hamiltonian cycle (path). Two facets of P will be called *neighbouring*, if they share a common edge. A polytope or one of its facets is called *simple*, if each of its vertices lies on precisely three edges of the polytope. The 1-skeleton of a simple polyhedron is a *cubic* graph, i.e. a graph in which all vertices are *cubic*, i.e. have degree 3. We call a face of a planar graph *cubic*, if all of its boundary vertices are cubic.

P is called a *k-pyramid* [Zamfirescu and Zamfirescu, 2011], if there exist at most k pairwise disjoint simple facets $F_1, ..., F_k$, called *bases*, such that every other facet has some neighbouring base. For $k \geq 2$, we call P *belted* if some pair of bases has no common neighbouring facet. In particular, we say that a belted 2-pyramid with basic cycles C_1, C_2 (i.e., C_1 and C_2 are disjoint cycles that bound the bases) is *simply belted* if every vertex of the unique cycle disjoint from $C_1 \cup C_2$ in P is of degree 3. Denote the family of the 1-skeleta of (cubic, belted) k-pyramids by $\mathscr{P}_k$ ($\mathscr{P}_k^3$, $\overline{\mathscr{P}}_k$), and write $\bigcup_k \mathscr{P}_k = \mathscr{P}$. ($\mathscr{P}^3$ and $\overline{\mathscr{P}}$ are defined analogously.) Notice that $\mathscr{P}_1$ are precisely the well-known Halin graphs. It was proven by Bondy [1975] that they are all hamiltonian – even 1-hamiltonian, i.e. they are hamiltonian and remain hamiltonian when removing an arbitrary vertex.

Consider $G \in \mathscr{P}_k$. G is a graph with at most k pairwise disjoint cycles called *basic cycles* which bound the bases. For $G \in \mathscr{P}$, define $k^* = k^*(G) = \min\{k \in \mathbb{N} : G \in \mathscr{P}_k\}$. For $G \notin \mathscr{P}$, put $k^*(G) = \infty$. Call such a set of (k^*) k basic cycles a (*minimal*) *foundation*. Notice that two foundations of the same graph may have different cardinalities, as for the n-gonal prism, $n \geq 6$. Also, observe that minimal foundations can be unique (see Fig. 1), but need not be (see Figs. 3 and 4); two minimal foundations may even have no basic cycle in common, as is the case for the graph in Fig. 3, or the hexagonal prism. Keep in mind that G need not be cubic, but all vertices on the basic cycles must have degree 3, i.e. their corresponding facets must be simple.

The *dual graph* G^* of a planar graph G has vertices each of which corresponds to a face of G, and each of whose faces corresponds to a graph vertex of G^*. Two vertices in G^* are connected by an edge iff the corresponding faces in G are adjacent. We shall write $g(F) = v$, if F is a face of G, and v its corresponding vertex in G^*.

Now let G be a polyhedral graph. We will write $G = (V(G), E(G))$ with $|V(G)| = n$ and $|E(G)| = m$, and put $G^* = (V(G^*), E(G^*))$, where $V(G^*) = \{v_1, ..., v_f\}$. For a vertex $v \in V(G)$ we denote by $N(v)$ the set of all neighbours of v, and put $N[v] = N(v) \cup \{v\}$. Additionally, let $\mathscr{F}$ ($\mathscr{F}^3$) be the set of all (cubic) faces of G. We have $|\mathscr{F}| = f$ and put $|\mathscr{F}^3| = r$.

An *independent set* $X \subset V(G)$ of a graph G contains exclusively vertices that are not pairwise adjacent. A set $Y \subseteq V(G)$ is called *dominating*, if every vertex of G is either in Y, or a neighbour of a vertex in Y. We denote the minimal cardinality of an independent dominating set in G by $i(G)$.

3 Computing k^* and testing beltedness

Let G be an arbitrary graph. For G to be polyhedral it must be planar and 3-connected. We present in the following two algorithms. The first (see Section 3.1) covers the case when the input graph is known to be cubic, whereas the second (see Section 3.2) deals with arbitrary input graphs. The purpose of these algorithms is not to showcase optimal run-times, but to demonstrate that the algorithmic problem has polynomial complexity. We output k^*, the minimal k for which $G \in \mathscr{P}_k$ (lower and upper bounds for k are provided in Appendix A). If $G \notin \mathscr{P}$, we output $k^* = \infty$.

Test 1: Planarity

We test the planarity of G using the Boyer-Myrvold planarity test and embedding algorithm (BMEA) [Boyer and Myrvold, 2004], which has a complexity of $O(n)$. Notice that after the embedding we can traverse the $f = 2+m-n \leq 2n-4$ faces of $\mathscr{F}$ in $O(n)$ time.

Test 2: 3-Connectedness

For any pair of vertices u, v in G we test whether $G \setminus \{u, v\}$ is connected. To verify whether a graph is connected or not can be done, for example, with a breadth-first search, which runs in $O(n+m)$ time. Accordingly, this processing step has a time complexity of $O(n^3)$ for planar graphs.

In the following we assume that the considered graph passed these two initial tests and, in consequence, is polyhedral.

Test 3: 3-Regularity

We determine whether the input graph is cubic in $O(n)$ time. If the graph is cubic, we apply the specialized algorithm described in Section 3.1, otherwise we proceed as described in Section 3.2.

3.1 Simple polyhedra

In this instance of the problem, the input graph G is given to be cubic, so by Euler's formula we have $f = r = n/2+2$. (P1) is automatically satisfied for cubic graphs due to the following short argument. Take an arbitrary face $F_1 \in \mathscr{F}$. If F_1 does not neighbour all other faces, take a second face $F_2 \in \mathscr{F}$, with $F_1 \cap F_2 = \emptyset$ (such an F_2 exists, as there are faces which do not neighbour F_1, and G is cubic). Check whether all faces have as neighbour F_1 or F_2. If this is the case, G is a 2-pyramid – if not, we may add a third face (as before), and so forth. Thus, in a cubic polyhedral graph we are always able to find a foundation, so there necessarily exists a k for which $G \in \mathscr{P}_k$.

We now address (P2) algorithmically. The basic idea of the following algorithm is to compute $i(G^*)$. Notice that in cubic graphs, MIDS and minimal foundations are in 1-to-1 correspondence, so for $G \in \mathscr{P}^3$ we have $i(G^*) = k^*(G)$. For the general polyhedral case, this need not be true, see Section 3.2.

Algorithm 1

Step 1

Denote by d_i the degree of the vertex $v_i \in V(G^*)$, $d_1 \geq \ldots \geq d_f$. We first verify whether $G \in \mathscr{P}_1^3$, which is equivalent to determining whether $d_1 \geq f-1$ holds or not. If this is the case, then G is the 1-skeleton of a 1-pyramid with base $g^{-1}(v_1)$, we output $k^* = 1$ and we stop here. Else, we have $k^* \geq 2$ and we continue with Step 2. Obviously, determining d_1 requires only $O(m) = O(n)$ time. Please note that f is a constant that can be calculated during the construction of the embedding as well as of G^* (the construction of the latter has a complexity of $O(n \log n)$, see Appendix A).

Step 2

Now we verify whether $G \in \mathscr{P}_2^3$ in a similar fashion. Put $N[v_i] = N_i$. We have to check for all $i, j \in \{1, \ldots, f\}$, $i < j$ (since all operations are commutative, this ordering does not change the result, albeit it does change the number of operations), the inequality

$$|N_i| + |N_j| - |N_i \cap N_j| \geq n/2 + 2.$$

If the above inequality holds, we output the minimal foundation $\{g^{-1}(v_i), g^{-1}(v_j)\}$, $k^* = 2$ (due to Step 1 we know that $k^* \neq 1$) and stop. Else, we have $k^* \geq 3$, and continue with Step 3.

Using, e.g., sorted adjacency lists to represent $N(v)$ and $N[v]$, which can be constructed in $O(n^2 \log n)$, we can calculate $N_i \cap N_j$ in $O(n)$ by applying set intersection algorithms for ordered data structures, see Knuth [1997]; Cormen et al. [2009]. Consequently, for each index pair i, j each term can be calculated with at most $O(2n-1) = O(n)$ operations. Thus, we obtain a runtime of

$$O\left(\frac{(2n-1)n(n-1)}{2}\right) = O\left(n^3\right).$$

Step 3

We proceed as above and verify for all $i, j, k \in \{1, \ldots, f\}$, $i < j < k$, the inequality

$$|N_i| + |N_j| + |N_k| - (|N_i \cap N_j| + |N_j \cap N_k| + |N_k \cap N_i|) + |N_i \cap N_j \cap N_k| \geq n/2 + 2.$$

Once more, if the above inequality holds, we output $k^* = 3$ and stop. Else, we have $k^* \geq 4$, and continue with Step 4.

As in Step 2, for all $O(n^3)$ index triples i, j, k we can calculate each term in $O(n)$ time. Furthermore, for fixed, small k the number of terms can be considered a quasi-constant (see Step 4 for arbitrary k). Thus, the decision whether or not a graph is a k-pyramid, $k \leq 3$, has a complexity of $O(n^4)$.

Step 4

If we have determined that $k^* \geq 4$, we can naturally generalize the approach from Steps 1 through 3 and apply the inclusion-exclusion principle. Accordingly, we give the following characterization of simple k-pyramids.

$$G \in \mathscr{P}_k^3 \iff \exists K \subset \{1, \ldots, f\} \text{ s.t. } |K| = k \text{ and } \sum_{I \subset K} (-1)^{|I|+1} \cdot \left| \bigcap_{i \in I} N_i \right| \geq n/2 + 2.$$

Again, using sorted adjacency lists, we can iteratively calculate each term (i.e. intersection) in $O(n)$ time. However, for arbitrary k we cannot consider the number of terms a quasi-constant and must take into account that the number of terms grows exponentially. For each subset K we have $\binom{f}{k}$ sums, each of which consists of 2^k terms, thus leading to a total of $\binom{f}{k}2^k$ terms and a total complexity of

$$O\left(n\binom{f}{k}2^k\right) = O\left(n\binom{2n-4}{k}2^k\right).$$

In light of the fact that [Zamfirescu and Zamfirescu, 2011] only provides applicable Theorems for $k \leq 3$, we may choose to stop the algorithm if it decided that $k^* \geq 4$. Notice that we determine whether k^* is 1, 2 or 3 in $O(n^4)$ time. However, in general, the number of terms and consequently the asymptotic runtime of the presented algorithm grows exponentially with k as well as n, which could be expected, because the related problem, i.e. MIDS, is NP-complete.

For (P3), please see Section 3.3.

3.2 Arbitrary polyhedra

Here, we must first address the question whether there exists a natural number k such that G is a k-pyramid.

Algorithm 2

Step 1

We begin by treating (P1) and (P2). Recall that $\mathscr{F}^3 = \{F_1, ..., F_r\} \subset \mathscr{F}$, with $g(F_i) = w_i$ and $\{w_i\}_{i=1}^r = W \subset V(G^*)$. There are several special cases:

(i) If $r = 0$, i.e. there are no cubic faces, we may stop here, and output that there is no k such that G is isomorphic to (the 1-skeleton of) a k-pyramid, i.e. $k^* = \infty$.

(ii) If $r \geq f - 2$, i.e. all faces are cubic, then G is cubic (and definitely a k-pyramid for some k) and we continue with Algorithm 1 from Section 3.1.

(iii) Every face must have at least two neighbouring vertices of degree 3 on its boundary. If this is not the case, G cannot be isomorphic to a k-pyramid, which we output as $k^* = \infty$.

Please note that since we can traverse all faces in $O(n)$ and all corresponding nodes of each face in $O(n)$ as well, the runtime of the steps above is trivially bounded by $O(n^2)$.

If none of the above occurs, we continue with Step 2.

Step 2

In contrast to the cubic case, put $N[w_i] = N_i$, and we have

$$G \in \mathscr{P}_k \iff \exists K \subset \{1, ..., r\} : |K| = k \text{ and } \sum_{I \subset K} (-1)^{|I|+1} \cdot \left| \bigcap_{i \in I} N_i \right| \geq n/2 + 2.$$

As in Algorithm 1, we search in G for an independent dominating set of minimal cardinality D, and we first verify whether $|D| = i(G^*) = 1$, then whether $|D| = i(G^*) = 2$, etc. Notice that we have the additional condition $D \subset W$. If such a set D is found, we output that $G \in \mathscr{P}_{|D|}$, and we have $|D| = k^*$. If no such set can be found, we output $k^* = \infty$.

Accordingly, we get the corresponding complexity of

$$O\left(n \binom{r}{k} 2^k\right) = O\left(n \binom{f}{k} 2^k\right),$$

see Step 4 of Algorithm 1.

3.3 Testing Non-Beltedness

Assume that for an input graph G we have determined with the Algorithms above that $G \in \mathscr{P}_{k^*}$. We can now treat (P3). Let $\mathfrak{C}$ be the set of all minimal foundations of G, and denote by d_G the shortest path metric on G. Algorithms 1 and 2 are constructive, so we know of at least one minimal foundation. Theoretically, due to the nature of the Theorems from Zamfirescu and Zamfirescu [2011], we are interested to know in which cases G is non-belted. Clearly,

$$G \text{ is not belted } \iff \exists \mathscr{C} \in \mathfrak{C}\ \forall B_1, B_2 \in \mathscr{C} : d_{G^*}(g(B_1), g(B_2)) = 2,$$

which we rewrite as the algorithmically more useful characterization

$$G \in \overline{\mathscr{P}} \iff \forall \mathscr{C} \in \mathfrak{C}\ \exists B_1, B_2 \in \mathscr{C} : d_{G^*}(g(B_1), g(B_2)) \geq 3.$$

This allows us to efficiently test the beltedness of G by going through all pairs of bases B_1, B_2 (the number of pairs is trivially bounded above by $\binom{f}{2}$) of an arbitrary minimal foundation, constructed during Algorithm 1 or 2, and computing $d_{G^*}(g(B_1), g(B_2))$ in the dual graph G^*. If none of these distances exceeds 2, then the k-pyramid is non-belted.

Naturally, testing for each of the $O(\binom{f}{2}) = O(n^2)$ pairs of bases B_1 and B_2 whether or not it has a distance of at least 3 can be implemented using breadth-first-search on the dual graph with complexity $O(n^2)$. Thus, the resulting total complexity is $O(n^4)$.

4 Applications: Small *k*-Pyramids

As first application of the algorithms presented in Section 3, we are now able to decide whether a given graph is a k-pyramid, and thereafter reliably compute k^*. Let us recall several Theorems from [Zamfirescu and Zamfirescu, 2011] which will be of great use.

Theorem A. *Every non-belted* 2*-pyramid is hamiltonian.*
Theorem B. *Every simply belted* 2*-pyramid is hamiltonian.*
Corollary. *Every simple* 2*-pyramid is hamiltonian.*
Theorem C. *Every* 2*-pyramid is traceable, but not necessarily hamiltonian.*
Theorem D. *Every non-belted* 3*-pyramid is traceable.*
Theorem E. *There exist simple non-hamiltonian* 3*-pyramids.*

We present three applications of our algorithms. We give bounds for k-pyramids with interesting properties concerning traceability and hamiltonicity. For a summarized view of all known lower and upper bounds in various situations, please see Section 5.

4.1 Non-traceability

Let G be a non-traceable k-pyramid. By Theorem C, we have $k \geq 3$. In the non-belted case, we have $k \geq 4$ due to Theorem D.

Theorem 1. *There exist non-traceable 5-pyramids.*

Proof. In Fig. 1 we present a non-traceable (belted) 5-pyramid G, and recall that there is no non-traceable k-pyramid known for $k \leq 4$. G is based on the well-known Herschel graph. The algorithm from Section 3.2 determined the foundation shown in Fig. 1. In this particular case it is easy to see that it is of minimal cardinality, as it corresponds to the unique independent dominating set of G^*.

Let us prove the non-traceability of G. Contracting the basic cycles (which in this case are exclusively triangles) to vertices yields a graph G' with the property that G is non-traceable iff G' is non-traceable. But G' is precisely a well-known variant of Herschel's graph, which is a non-traceable polyhedral graph. □

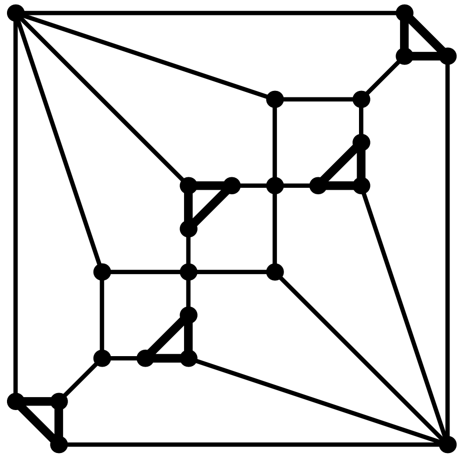
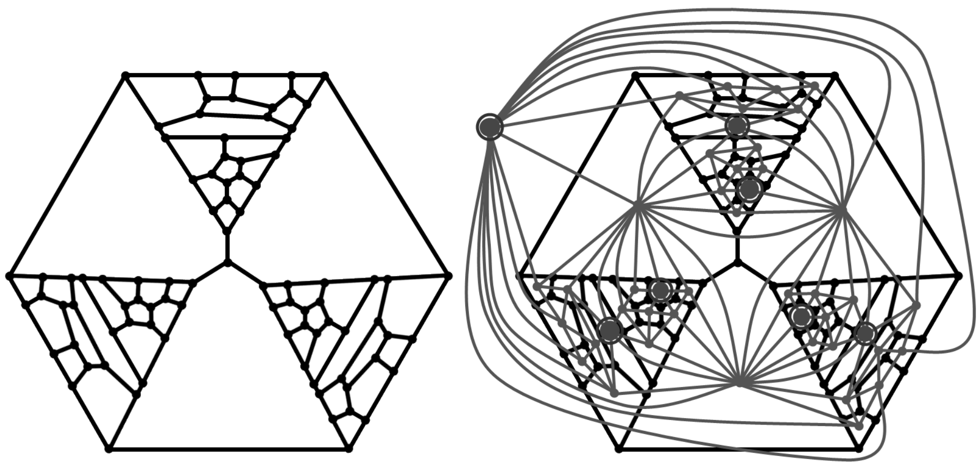


Fig. 1: A non-traceable 5-pyramid, and its unique minimal foundation marked by bold edges.

If we additionally require G to be cubic, we have the following.

Theorem 2. *There exist simple non-traceable* 7*-pyramids.*

Proof. The graph G from Fig. 2 (a) is non-traceable, see [Zamfirescu, 1980]. Fig. 2 (b) shows G and a MIDS of G^*, D, which corresponds to a minimal foundation in G.



(a) (b)

Fig. 2: (a) shows a non-traceable simple 7-pyramid G. (b) depicts G, its dual G^*, and as bold dots a MIDS of G^*.

As $|D| = 7$ and Algorithm 1 confirms that there exists no foundation of G of cardinality less than 7, G is a 7-pyramid. □

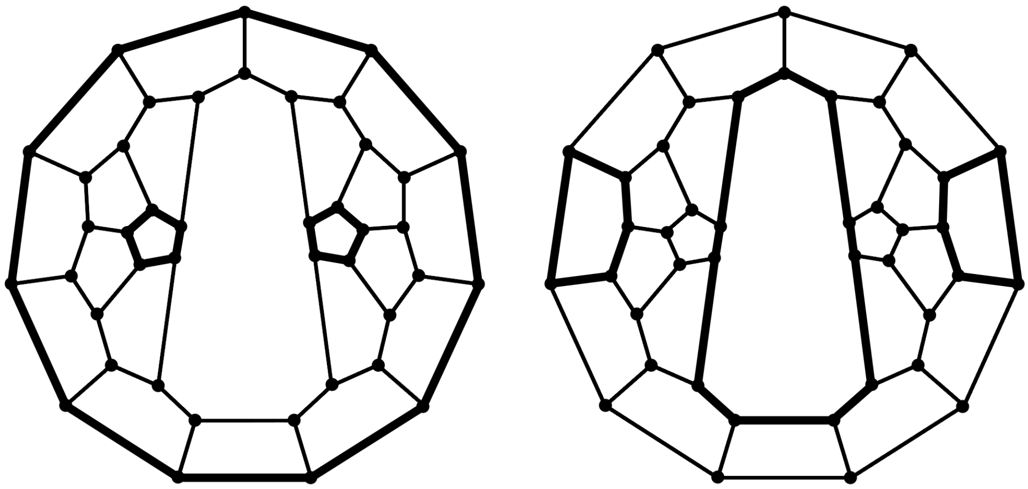
4.2 Non-hamiltonicity

In this Section, all graphs will be non-hamiltonian. For k-pyramids in general, we have the following. [Zamfirescu and Zamfirescu, 2011, Fig. 9] presents a 2-pyramid (1-pyramids are hamiltonian [Bondy, 1975]); this also covers the belted case. In the cubic case, the Corollary and Theorem E, plus the graph from Fig. 2, yield $k = 3$. For non-belted k-pyramids (simple or not), we only have a bound given by Theorem A, namely $k \geq 3$. For belted simple k-pyramids, we have $k = 3$ due to the graph from Fig. 4 and the Corollary.

Theorem 3. *All simple* 3*-pyramids on 38 or fewer vertices are hamiltonian. Furthermore, on 42 vertices there exist non-hamiltonian simple* 3*-pyramids.*

Proof. It is a classical result [Holton and McKay, 1989] that all cubic polyhedral graphs on 36 or fewer vertices are hamiltonian. On one hand, there exist exactly six cubic polyhedral non-hamiltonian graphs of order 38, see [Aldred et al., 2000, Fig. 1], and we omit here the straightforward verification using Algorithm 1 that indeed none of these six graphs is a k-pyramid for $k \leq 3$. On the other hand, there are many non-hamiltonian simple 3-pyramids on 42 vertices [Aldred et al., 2000]. One such graph is depicted in Fig. 3. □

It remains to be settled whether there exist simple non-hamiltonian 3-pyramids on 40 vertices.



(a) (b)

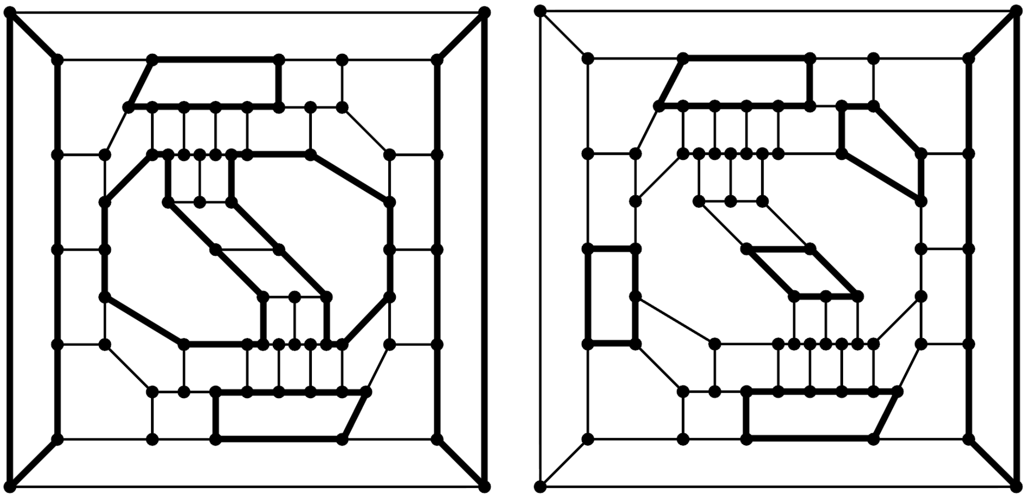
Fig. 3: The graph NH42.b from [Aldred et al., 2000, Fig. 2] on 42 vertices, also known as Grinberg's graph, with (a) and (b) showing two distinct minimal foundations.

4.3 Hypohamiltonicity

A graph is called *hypohamiltonian*, if it is non-hamiltonian, but by deleting an arbitrary vertex it becomes hamiltonian. Here we briefly discuss a special case of the results which can be found in Section 4.2, as every hypohamiltonian graph is non-hamiltonian.

Theorem 4. *There exist hypohamiltonian* 6*-pyramids.*

Proof. Fig. 4 shows a hypohamiltonian 6-pyramid (see [Wiener and Araya, 2011]), which has 70 vertices.



(a) (b)

Fig. 4: The Wiener-Araya graph [Wiener and Araya, 2011, Fig. 1] on 70 vertices is a hypohamiltonian 6-pyramid. (a) and (b) show two distinct minimal foundations.

The proof that the graph from Fig. 4 is indeed hypohamiltonian can be found in [Wiener and Araya, 2011]. The two minimal foundations shown in Fig. 4 (bounded by bold edges) were found using Algorithm 1, see Section 3.1. □

5 Discussion

The following table shows lower and upper bounds for the minimal k for which there exists a k-pyramid with the mentioned attributes. The lower bounds can be derived from Theorems A through E and the Corollary, and the upper bounds from Section 4. Where no upper bounds are given, there may not exist a k satisfying the imposed conditions.

	non-traceable		non-hamiltonian		hypohamiltonian	
	non-belted	belted	non-belted	belted	non-belted	belted
polyhedral	$k \geq 4$	$3 \leq k \leq 5$	$k \geq 3$	$k = 2$	$k \geq 3$	$2 \leq k \leq 6$
cubic	$k \geq 4$	$k \geq 3$	$k \geq 3$	$k = 3$	$k \geq 3$	$3 \leq k \leq 6$

6 Open Problems

Several interesting questions remain unanswered. We select the following.

1. Is every 3-pyramid traceable?

2. Is there an efficient way to count (minimal) foundations?

2. Is deciding whether a given polyhedron is a k-pyramid an NP-complete problem?

A Bounds

A.1 Lower Bound

Using the embedding provided by the BMEA, we compute the dual graph G^*, which can be done in $O(m \log m) = O(n \log n)$ time (cf. Kahng et al. [2008]).

Firstly, list all cubic faces $\mathscr{F}^3 = \{F_1, ..., F_r\}$, and put $\{w_1, ..., w_r\} = W$, where $g(F_i) = w_i$, $1 \leq i \leq r$. In order to do this in $O(n \cdot f) = O(n^2)$ time, we proceed as follows. Firstly, each vertex is marked with its degree – actually, this is done during graph construction; however, even when done separately the complexity is at most $O(n^2)$, depending on the basic graph data structure. Secondly, each vertex is marked with the adjacent faces, which is done during the BMEA. This implies a runtime of at most $O(n \cdot f)$ to cycle through all faces and their respective adjacent vertices. We remark that during this procedure, the boundary vertices of all faces are stored in an ordered manner – this will be of use later, but implies additional costs for sorting; thus, we practically get a complexity of $O((n \log n) \cdot f)$. Now, if G possesses a foundation $\mathscr{C}$ (for cubic graphs this is guaranteed, see 3.1), by definition $\mathscr{C} \subset \mathscr{F}^3$.

Compute the degrees $\{\deg(w_i) = d_i\}_{i=1}^r$ of the vertices of G^*, with $d_1 \geq ... \geq d_r$. We then have

$$G \in \mathscr{P}_k \implies \sum_{i=1}^{k} d_i + k \geq f.$$

This is a useful test in order to determine whether G has any chance of being a k-pyramid or not. This step also provides a set of candidate vertices in G^*, the associated faces of which may form a foundation of G.

A.2 Upper Bound

We have $k \leq f/4$, as every base has at least three neighbouring facets.

References

Aldred, R. E. L., Bau, S., Holton, D. A., and McKay, B. D. Nonhamiltonian 3-connected cubic planar graphs. *SIAM J. Discrete Math.*, 13(1):25–32, 2000.

Bondy, J. A. Pancyclic graphs: recent results. *Colloq. Math. Soc. János Bolyai*, 10: 181–187, 1975.

Boyer, J. M. and Myrvold, W. J. On the cutting edge: simplified $O(n)$ planarity by edge addition. *J. Graph Algorithms Appl.*, 8(3):241–273, 2004.

Brandstädt, A., Chepoi, V. D., and Dragan, F. F. The algorithmic use of hypertree structure and maximum neighbourhood orderings. *Discrete Appl. Math.*, 82(1-3): 43–77, 1998.

Broersma, H. J., Kloks, T., Kratsch, D., and Müller, H. Independent sets in asteroidal triple-free graphs. In *Proc. 24th Int. Colloq. Automata, Languages and Programming, LNCS (Springer)*, volume 1256, pages 760–770, 1997.

Chang, M. S. Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM J. Comput.*, 27(6):1671–1694, 1998.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms*. MIT Press and McGraw-Hill, third edition edition, 2009.

Corneil, D. G. and Perl, Y. Clustering and domination in perfect graphs. *Discrete Appl. Math.*, 9:27–40, 1984.

Farber, M. Independent domination in chordal graphs. *Oper. Res. Lett.*, 1:134–138, 1982.

Garey, M. R. and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

Garey, M. R., Johnson, D. S., and Tarjan, R. E. The planar hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5:704–714, 1976.

Gaspers, S. and Liedloff, M. A branch-and-reduce algorithm for finding a minimum independent dominating set in graphs. *Technical Report no. 344, Univ. Bergen*, 2007.

Grinstead, D. L. and Slater, P. J. A recurrence template for several parameters in series-parallel graphs. *Discrete Appl. Math.*, 54:151–168, 1994.

Holton, D. A. and McKay, B. D. The smallest non-hamiltonian 3-connected cubic planar graphs have 38 vertices. *J. Combin. Theory, Ser. B*, 45:305–319, 1989.

Irving, R. W. On approximating the minimum independent dominating set. *Inf. Proc. Lett.*, 37:197–200, 1991.

Kahng, A. B., Park, C.-H., and Xu, X. Fast dual-graph-based hotspot filtering. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 27(9), 2008.

Karp, R. M. *Reducibility Among Combinatorial Problems*, pages 85–103. Plenum Press, New York, 1972.

Knuth, D. *The Art of Computer Programming*, volume 3. Addison-Wesley, 1997. Sorting and Searching; Section 5.2.4: Sorting by Merging. Section 5.3.2: Minimum-Comparison Merging.

Kratsch, D. and Stewart, L. Domination on cocomparability graphs. *SIAM J. Discrete Math.*, 6:400–417, 1993.

Liu, C. and Song, Y. Exact algorithms for finding the minimum independent dominating set in graphs. In *ISAAC 2006, in: LNCS (Springer)*, volume 4288, pages 439–448, 2006.

Malik, S., Qureshi, A. M., and Zamfirescu, T. Hamiltonian properties of generalized Halin graphs. *Can. Math. Bull.*, 52(3):416–423, 2009.

Manlove, D. F. On the algorithmic complexity of twelve covering and independence parameters of graphs. *Discrete Appl. Math.*, 91(1–3):155–175, 1999.

Pfaff, J., Laskar, R., and Hedetniemi, S. T. Linear algorithms for independent domination and total domination in series-parallel graphs. *Congressus Numerantium*, 45:71–82, 1984.

Skowrońska, M. Hamiltonian properties of Halin-like graphs. *Ars Combin.*, 16(B): 97–109, 1983.

Skowrońska, M. and Sysło, M. M. Hamiltonian cycles in skirted trees. *Zastosow. Mat.*, 19(3–4):599–610, 1987.

Skupień, Z. *Crowned trees and planar highly hamiltonian graphs*, pages 537–555. Wissenschaftsverlag Mannheim, 1990.

Steinitz, E. *Polyeder und Raumeinteilungen*, volume 3 (Geometrie), pages 1–139. 1922.

Wiener, G. and Araya, M. On cubic planar hypohamiltonian and hypotraceable graphs. *Electr. J. Combin.*, # P85:1–11, 2011.

Yannakakis, M. and Gavril, F. Edge dominating sets in graphs. *SIAM J. Appl. Math.*, 38:364–372, 1980.

Zamfirescu, C. T. and Zamfirescu, T. I. Hamiltonian properties of generalized pyramids. *Math. Nachr.*, 284(13):1739–1747, 2011.

Zamfirescu, T. Three small cubic graphs with interesting hamiltonian properties. *J. Graph Theory*, 4:287–292, 1980.

BORIS SCHAUERTE

Institut für Anthropomatik und Robotik,
Karlsruhe Institut für Technologie,
Germany

E-mail address: *boris.schauerte@kit.edu*

CAROL T. ZAMFIRESCU

Fakultät für Mathematik,
Technische Universität Dortmund,
Germany

E-mail address: *czamfirescu@gmail.com*