On the complexity of recognizing Hamming graphs and related classes of graphs

Wilfried Imrich
Department of Mathematics and Applied Geometry
Montanuniversität Leoben
A-8700 Leoben, Austria

Sandi Klavžar *
Department of Mathematics, PEF
University of Maribor
Koroška cesta 160
62000 Maribor
Slovenia

Abstract

This article contains a new algorithm that recognizes whether a given graph G is a Hamming graph, i.e. a Cartesian product of complete graphs, in $O(m)$ time and $O(n^2)$ space. Here m and n denote the numbers of edges and vertices of G, respectively. Previously this was only possible in $O(m \log n)$ time.

Moreover, we present a survey of other recognition algorithms for Hamming graphs, retracts of Hamming graphs and isometric subgraphs of Hamming graphs. Special emphasis is also given to the bipartite case in which these classes are reduced to binary Hamming graphs, median graphs and partial binary Hamming graphs.

1 Introduction

This paper is a contribution to the recognition of classes of graphs defined by metric properties. These classes include Hamming graphs, quasi-median graphs, partial Hamming graphs, binary Hamming graphs, median graphs and partial binary Hamming graphs.

We shall define the above mentioned classes of graphs, list some of their structural properties, in particular those which are exploited from the algorithmic point of view, and then shortly describe the ideas behind several recognition algorithms.

Moreover, we also present a new recognition algorithm which is optimal in its time complexity for the recognition of Hamming graphs.

All graphs considered in this paper are finite undirected graphs without loops or multiple edges. Throughout the paper, for a given graph G, let n and m stand for the number

*This work was supported in part by the Ministry of Science and Technology of Slovenia under the grant J1-7036.

of its vertices and edges, respectively. For a graph G and a vertex set $X \subset V(G)$ let $\langle X \rangle$ denote the subgraph of G induced by X.

A subgraph H of a graph G is an *isometric* subgraph, if $d_H(u,v) = d_G(u,v)$ for all $u, v \in V(H)$. In addition, if $\alpha : V(H) \to V(G)$ maps edges into edges and if $\alpha(H)$ is an isometric subgraph of G, we call α an *isometric embedding* of H into G.

The *interval* $I(u,v)$ between vertices u and v consists of all vertices on shortest paths between u and v. A subgraph H of G is *convex*, if for any u, $v \in V(H)$, $I(u,v) \subseteq V(H)$. Clearly a convex subgraph is an isometric subgraph but the converse need not be true. The *convex hull* of a subgraph H in G is the intersection of all convex subgraphs of G containing H, i.e., the smallest convex subgraph containing H.

The *Cartesian product* $G \,\square\, H$ of graphs G and H is the graph with vertex set $V(G) \times V(H)$ and $(a,x)(b,y) \in E(G \,\square\, H)$ whenever $ab \in E(G)$ and $x = y$, or $a = b$ and $xy \in E(H)$.

A mapping $f : V(G) \to V(H)$ is a graph *homomorphism* if $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$. A subgraph H of a graph G is a *retract* of G, if there is a homomorphism r from $V(G)$ to $V(H)$ such that $r(v) = v$ for every $v \in V(H)$. The map r is called a *retraction.* If we allow that r maps an edge of G either to an edge or to a single vertex in H, we call H a *weak retract* of G and r a *weak retraction.*

2 The bipartite or binary case

In this section we consider binary Hamming graphs, their retracts and isometric subgraphs.

2.1 Binary Hamming graphs

Binary Hamming graphs are also known as hypercubes. A d-dimensional *hypercube* Q_d (*d-cube* for short) is a graph whose vertices are all d-tuples $b_1 b_2 \ldots b_d$ with $b_i \in \{0,1\}$ and two vertices are adjacent if the corresponding tuples differ in precisely one coordinate. Alternatively, Q_d is the Cartesian product of d copies of K_2. Clearly Q_d is a connected, bipartite d-regular graph on $n = 2^d$ vertices and $m = d \cdot 2^{d-1}$ edges. Moreover, its automorphism group is vertex and edge transitive. We also note that the usual shortest path distance between any two vertices x and y of Q_d is the number of positions in which x and y differ. For example, the distance between 0110 and 1101 is 3. This distance is also called the *Hamming distance* between x and y.

Bhat [9] proposed an $O(m)$ algorithm for recognizing binary Hamming graphs. In fact, his algorithm can essentially be obtained by specializing the algorithm for Hamming graphs which we will present in Section 3.1. Here we propose an alternative algorithm of the same time complexity for recognizing binary Hamming graphs. One can obtain it by applying an algorithm from [26] or an algorithm from [22] for recognizing median graphs. The time complexities of these two algorithms, which will be discussed in the next section, can be reduced for binary Hamming graphs because we need not worry about convexity.

Let uv be an arbitrary edge of Q_d. To fix ideas, let $u = 00 \ldots 0$ and $v = 10 \ldots 0$. Then the vertices whose first coordinate is zero are exactly those vertices V_{uv} in Q_d which are closer to u than to v. Furthermore, they induce a $(d-1)$-cube $\langle V_{uv} \rangle$ in Q_d. Analogously we obtain a $(d-1)$-cube $\langle V_{vu}\rangle$ on the set V_{vu} of vertices that are closer to v than to u, i.e. those vertices the first coordinate of which is 1.

The edges not in $\langle V_{uv}\rangle$ or $\langle V_{vu}\rangle$ are of the form

$$(0x_2x_3 \ldots x_d)(1x_2x_3 \ldots x_d).$$

It is easily seen that these edges are a matching of Q_d and that this matching defines an isomorphism

$$\alpha : 0x_2x_3 \ldots x_d \mapsto 1x_2x_3 \ldots x_d$$

of $\langle V_{uv}\rangle$ onto $\langle V_{vu}\rangle$.

This information already suffices for an $O(n \log n)$ algorithm for recognizing binary Hamming graphs. For, let G be a connected graph on n vertices. If it is a binary Hamming graph its number m of edges must be $\frac{n}{2}\log_2 n = O(n \log n)$. First, we can check in that many steps whether G is bipartite. Then, choosing $uv \in E(G)$ arbitrarily we can obtain all distances $d_G(u,x)$ and $d_G(v,x)$ for $x \in V(G)$ in $2m$ steps. Thus, V_{uv} and V_{vu} can be determined in $O(m)$ time. In another $O(m)$ steps one can determine whether the edges not in $\langle V_{uv}\rangle$ or $\langle V_{vu}\rangle$ define a matching and an isomorphism between $\langle V_{uv}\rangle$ and $\langle V_{vu}\rangle$.

It remains to show that $\langle V_{uv}\rangle$ which has less than $\frac{m}{2}$ edges is a binary Hamming graph. The complexity thus is

$$O(m) + O(\frac{m}{2}) + O(\frac{m}{4}) + \cdots + O(1) = O(m).$$

Since all edges have to be checked this complexity is best possible. We formulate this as a theorem.

Theorem 2.1. **[9]** *For a given graph G on n vertices one can decide in $O(n \log n)$ steps whether G is a binary Hamming graph. This complexity is optimal.*

For the next two sections we recall that for any subgraph of a binary Hamming graph, $m \leq \frac{1}{2}n \log n$, cf. [2, 18].

2.2 Median graphs

A *median* of a set of three vertices u, v and w is a vertex that lies in $I(u,v) \cap I(u,w) \cap I(v,w)$. In other words, x is a median of u, v and w if

$$d(u,x) + d(x,v) = d(u,v),$$
$$d(v,x) + d(x,w) = d(v,w),$$
$$d(u,x) + d(x,w) = d(u,w).$$

Let $uv \in E(G)$ and let $w \in V(G)$. It is easy to verify that u, v and w have a median if and only if $d(u,w) \neq d(v,w)$. This observation in turn implies that G is bipartite if any three vertices of a graph G have a median.

A connected graph G is a *median graph* if every triple of its vertices has a unique median. It is easily seen that binary Hamming graphs and trees are median graphs. Furthermore, Bandelt proved the following theorem.

Theorem 2.2. **[6]** *A graph is a median graph if and only if it is a retract of a binary Hamming graphs.*

In fact, one could also replace retracts by weak retracts in Theorem 2.2. We also note that a retract always is an isometric subgraph.

Next, we describe the convex expansion procedure due to Mulder [27, 28] which leads to another characterization of median graphs. Let G be a graph. Furthermore, suppose $W \subseteq V(G)$ and $W' \subseteq V(G)$ are vertex sets such that $W \cup W' = V(G)$, $W \cap W' \neq \emptyset$ and there is no edge between $W \setminus W'$ and $W' \setminus W$. The *expansion* of G with respect to W and W' is the graph H constructed as follows:

(i) replace every vertex $v \in W \cap W'$ by an edge $u_v u'_v$,
(ii) join u_v to the neighbors of v in $W \setminus W'$ and u'_v to the neighbors of v in $W' \setminus W$,
(iii) for adjacent vertices $v, w \in W \cap W'$, join u_v to u_w and u'_v to u'_w.

If in addition, $\langle W \rangle$ and $\langle W' \rangle$ are convex subgraphs of G, H is a *convex expansion* of G. Mulder proved the following important result.

Theorem 2.3. **[27]** *A graph is a median graph if and only if it can be obtained from K_1 by a sequence of convex expansions.*

We now turn to the algorithmic point of view. As a by-product of their investigation, Chung, Graham and Saks [11] proposed an $O(n^4)$ algorithm for recognizing median graphs.

Jha and Slutzki have given two algorithms of complexity $O(n^2 \log n)$. One [25] is based on Bandelt's characterization, the other one [26] uses the Mulder's convex expansion procedure. The main bottleneck of the latter approach from the computational point of view is a convexity test. This is partially solved by the following lemma due to Bandelt (personal communication to Jha and Slutzki). For a graph G call a subgraph H of G 2- *convex* if for any two vertices u and v of H with $d_G(u, v) = 2$, every common neighbor of u and v belongs to H.

Lemma 2.4. **[26]** *Let G be a connected bipartite graph in which every triple of vertices has a median. Then a subgraph H of G is convex if and only if H is a 2-convex, isometric subgraph of G.*

In fact, as pointed out by a referee, it is possible to replace isometric subgraphs by connected subgraphs in the formulation of Lemma 2.4.

The fastest known algorithm for recognizing median graphs, however, is due to Hagauer, Imrich and Klavžar [22]. It has time complexity $O(n^{\frac{3}{2}} \log n)$ and is also based on Mulder's convex expansion. The first part of the algorithm attempts to embed a given graph G isometrically into a hypercube. It properly embeds every median graph and rejects all non-embeddable graphs and some embeddable ones. (It can thus not be used as a recognition algorithm for partial binary Hamming graphs.) This first part has time complexity $O(m \log n)$. In a second step the convexity of certain subgraphs of G has to be tested. (These graphs correspond to the graphs $\langle W \rangle$ and $\langle W' \rangle$ introduced before Theorem 2.3. Their number can be of order $O(n)$.) If one performs these tests indiscriminately one by one, the complexity may go up to $O(mn)$. In [22] the sequence of these tests is carefully chosen and thus allows a reduction of the complexity to $O(mn^{\frac{1}{2}})$. We can thus state:

Theorem 2.5. **[22]** *For a given graph G on n vertices one can decide in $O(n^{\frac{3}{2}} \log n)$ steps whether G is a median graph.*

2.3 Partial binary Hamming graphs

Graphs that can be isometrically embedded into a binary Hamming graph are called *partial binary Hamming graphs*. In other words, a graph G is a partial binary Hamming graph if its vertices can be labelled by binary labels of a fixed length such that the distance between any two vertices of G is equal to the Hamming distance between the corresponding labels. As median graphs are partial binary Hamming graphs and as the cycle C_6 is a partial binary Hamming graph which is not a median graph, partial binary Hamming graphs form a proper extension of median graphs.

Considering the structure of partial binary Hamming graphs which could be useful for a fast recognition algorithm, the following relation plays a central role.

Let G be a connected graph. Define a relation Θ on $E(G)$ as follows. If $e = xy \in E(G)$ and $f = uv \in E(G)$, then $e \Theta f$ if

$$d(x,u) + d(y,v) \;\neq\; d(x,v) + d(y,u).$$

The relation Θ is reflexive and symmetric, yet it need not be transitive. We denote its transitive closure by Θ^*. Winkler proved the following result which is the base for a fast recognition algorithm.

Theorem 2.6. **[33]** *Let G be a connected graph. Then G is a partial binary Hamming graph if and only if G is bipartite and $\Theta^* = \Theta$.*

Before we continue we would like to mention that several other characterizations of partial binary Hamming graphs are known, the first one being due to Djoković [16]. Furthermore, Chepoi [10] has proved a similar result to Theorem 2.3 for partial binary Hamming graphs (one has to replace "convex expansion" with "isometric expansion").

Aurenhammer and Hagauer demonstrated in [3] how to compute the relation Θ^* in $O(nm)$ time. In [2] they used this result for deciding transitivity of Θ for bipartite graphs within the same time bound. The main idea is that one only counts the number of pairs of edges being in relation Θ and then compares this number with the number of pairs of edges in relation Θ^*. Since for a partial binary Hamming graph $m \leq \frac{1}{2} n \log n$ holds, this leads to an $O(n^2 \log n)$ algorithm for recognizing partial binary Hamming graphs.

A much simpler recognition algorithm of the same time complexity was proposed by Imrich and Klavžar [24]. Its main advantage is that one only needs to compute Θ^* and not Θ itself. Combining this idea with an approach of Feder [17] for computing Θ^* we obtained a simple algorithm for recognizing partial binary Hamming graphs. We shall present more details in Section 3.4 when the general case of partial Hamming graphs is treated. The algorithm for the binary case is a straightforward specialization of the general algorithm. We thus have:

Theorem 2.7. **[2, 24]** *For a given graph G on n vertices one can decide in $O(n^2 \log n)$ steps whether G is a partial binary Hamming graph.*

Main algorithmic results of Section 2 are summarized in Table 1.

class of graphs	time complexity
binary Hamming graphs	$n \log n$
median graphs	$n^{\frac{3}{2}} \log n$
partial binary Hamming graphs	$n^2 \log n$

Table 1: Complexities for the binary case

3 The general case

Binary Hamming graphs are Cartesian products of K_2's. A natural generalization are Cartesian products of arbitrary complete graphs. These products are known as Hamming graphs. As before we can consider retracts and isometric subgraphs and ask for properties and recognition algorithms. This approach yields several interesting classes of graphs.

3.1 Hamming graphs

A *Hamming graph* is the Cartesian product of complete graphs. Many characterizations of these graphs are known, we refer to [7, 8] and references there.

Suppose we wish to recognize Hamming graphs. Then, for a given graph it is enough to find its (prime) factor decomposition with respect to the Cartesian product and verify whether the factors are complete graphs. The fastest known algorithm for such a decomposition is due to Aurenhammer, Hagauer and Imrich [4] and is of time complexity $O(m \log n)$. Here we will reduce this complexity to $O(m)$ for the special case of Hamming graphs.

For our purposes the following definition will be convenient.

Let $r_1, r_2, \ldots, r_t$ be given integers ≥ 2 and let V be the set of t-tuples $a_1 a_2 \ldots a_t$ with $0 \leq a_i \leq r_t - 1$. These t-tuples will be the set of vertices of our Hamming graph. We note that there are $n = \prod_{i=1}^{t} r_i$ such t-tuples.

We connect any two t-tuples $a_1 a_2 \ldots a_t$ and $b_1 b_2 \ldots b_t$ by an edge if they differ in exactly one place, i.e. if there is a j such that $a_j \neq b_j$ but $a_i = b_i$ for $i \neq j$. Let E be the set of such edges. Then it is straightforward to see that the graph $H = (V, E)$ is a Hamming graph.

It is easy to see that the shortest path distance in H between any two vertices $a_1 a_2 \ldots a_t$ and $b_1 b_2 \ldots b_t$ is the number of places (or components) in which these t-tuples differ. This distance is also called the *Hamming distance* (cf. Section 2.1 for the bipartite case) and the corresponding labelling of the vertices of H a *Hamming labelling.*

Let $v_0 = 00 \ldots 0$ and let $v_0, v_1, \ldots, v_n$ be a BFS ordering of the vertices of H. Furthermore, let L_k denote the k-th level with respect to this ordering, i.e. the set of all vertices of distance k from v_0.

Clearly L_0 consists only of v_0 and L_1 of all neighbors of v_0. In general we can say that L_k consists of all those t-tuples $a_1 a_2 \ldots a_t$ in which exactly k of the a_i are $\neq 0$. For further reference we state the following observations as Facts.

Fact 3.1. *Let π be a permutation of* $\{0, 1, \ldots, r_i - 1\}$. *If*

$$h : v \mapsto a_1 a_2 \ldots a_i \ldots a_t$$

is a Hamming labelling of H, then

$$\pi h : v \mapsto a_1 a_2 \ldots \pi a_i \ldots a_t$$

is also a Hamming labelling.

Fact 3.2. *Let $1 \leq i < j \leq t$ and h be given as in Fact 3.1. Then*

$$h_{ij} : v \mapsto a_1 a_2 \ldots a_{i-1} a_j a_{i+1} \ldots a_{j-1} a_i a_{j+1} \ldots a_t$$

is also a Hamming labelling.

Fact 3.3. *The vertices of type $0 \ldots 0 a_i 0 \ldots 0, a_i \neq 0$, form a complete graph G_i on $r_i - 1$ vertices and there are no edges between G_i and G_j for $i \neq j$.*

Fact 3.4. *Let $u = a_1 a_2 \ldots a_t \in L_k$, $k \geq 1$. Then every neighbor v of u in L_{k-1} has exactly one more vanishing component than u.*

Also, if $k \geq 2$ the vertex u has at least two neighbors v, w in L_{k-1} and they differ in exactly two coordinates.

Moreover, if $v = b_1 b_2 \ldots b_t$ and $w = c_1 c_2 \ldots c_t$ then $a_i = \max\{b_i, c_i\}$ for $i = 1, \ldots, t$.

Suppose we are given a Hamming graph H by its adjacency matrix A. Then we can assign labels to its vertices by the following algorithm.

The Labelling Algorithm
Input: The adjacency matrix of a Hamming graph H.
Output: A Hamming labelling of H.

1. Choose a vertex v_0.

2. Arrange the vertices of H in levels $L_0, L_1, \ldots, L_k$ such that L_i contains all vertices in H of distance i from v_0.

3. Find the connected components of the subgraph of H spanned by the vertices in L_1. Let these components be $C_1, C_2, \ldots, C_t$ with $r_1 - 1, r_2 - 2, \ldots, r_t - 1$ vertices, respectively.

4. Label v_0 with a vector of length t containing only zeros.

5. Label the vertices of C_i with vectors of the form $0 \ldots 0 a_i 0 \ldots 0$, i.e. vectors of length t in which only the i-th coordinate a_i is different from zero, but where a_i assumes all values between 1 and $r_i - 1$.

6. Suppose all vertices in L_j, $1 \leq j < k$, have already been labelled. Choose an unlabelled vertex u in L_{j+1}. It must have at least two neighbors v, w in L_j. Let the labels of v and w be $b_1 b_2 \ldots b_t$ and $c_1 c_2 \ldots c_t$, respectively. Setting $a_i = \max\{b_i, c_i\}$ we obtain a label $a_1 a_2 \ldots a_t$ for u.

Proposition 3.5. *The labelling algorithm, applied to a Hamming graph H, yields a Hamming labelling of H.*

Proof. By Fact 3.1 there is a Hamming labelling of H where v_0 has the label $00 \ldots 0$.

By Fact 3.3 the labels of the vertices in L_1 have only one non-zero coordinate. Moreover, all vertices in a C_i differ in one and the same coordinate from v_0. By Fact 3.2 these coordinates can be arbitrarily assigned.

Once all vertices of L_1 are labelled, the labels of L_2 and all higher levels are determined by Fact 3.4. □

Proposition 3.6. *The time complexity of the labelling algorithm is $O(m)$ and the space complexity is $O(n^2)$.*

Proof. The space complexity is determined by the size of the adjacency matrix. This matrix is needed to be able to check in constant time whether edges between given endpoints exist.

We now investigate the time complexity of the algorithm.

Steps (1) and (4) require constant time.

Steps (2), (3) and (5) can each be completed in $O(m)$ time.

Neighbors $v, w \in L_j$ of $u \in L_{j+1}$ can be chosen in constant time and the new label for u can be formed in time $O(t)$. Let $n = |V(H)|$. Then the complexity of step (6) is $O(nt)$.

Since every vertex of H has at least t neighbors we infer $nt \leq 2m$. Hence, $O(nt) = O(m)$. □

Thus, for Hamming graphs H our labelling algorithm yields a Hamming labelling in $O(m)$ time. Given any graph G of which we wish to find out whether it is a Hamming graph or not, we can try to apply the labelling algorithm. If it cannot be completed G cannot be a Hamming graph. However, if it succeeds G still need not be a Hamming graph. Consider, for instance, a simple example from Fig. 1 where a (bipartite) non-Hamming graph G is presented together with a labelling obtained by the labelling algorithm. Note that this is the Hamming labelling of the 3-cube Q_3 and that G has the same number of vertices and edges as Q_3.

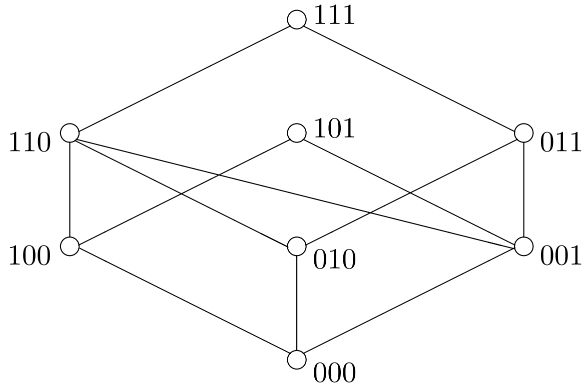

Figure 1: A non-Hamming graph G with a labelling

How and how fast can we check if a labelled graph is indeed a Hamming graph? We may assume that all labels of the form $a_1a_2 \ldots a_t$ with $0 \le a_i \le r_i - 1$ really occur, for otherwise G is not a Hamming graph. But then we can check in one run whether all edges which a Hamming graph with this labelling must have really occur. Every such check can be done in constant time since we work with the adjacency matrix. If no edges remain, G is a Hamming graph.

The Hamming Graph Algorithm
Input: The adjacency matrix A of a graph G.
Output: A Hamming labelling of G if it exists, rejection otherwise.

1. Choose a vertex v_0.
2. Arrange the vertices of G in levels $L_0, L_1, \ldots, L_k$ such that L_i contains all vertices in G of distance i from v_0.
3. Find the connected components of the subgraph of G spanned by the vertices of L_1. Let these components be $C_1, C_2, \ldots, C_t$ with $r_1 - 1, r_2 - 2, \ldots, r_t - 1$ vertices, respectively.
4. (a) If any of the subgraphs of G spanned by the C_i is not complete then reject.
   (b) If $n \neq \prod_{i=1}^{t} r_i$ then reject.
   (c) If $m \neq \frac{1}{2}\sum_{i=1}^{t}(r_i(r_i - 1)\prod_{j=1,j\neq i}^{t} r_j)$ then reject.
   (d) Form the vertices of the Hamming graph H with the labels $a_1a_2 \ldots a_t$ where $a_i \in \{0, 1, \ldots, r_i - 1\}$.
   (e) Label v_0 with a vector of length t containing only zeros.
5. Label the vertices of C_i with vectors of the form $0 \ldots 0a_i0 \ldots 0$, where $a_i \in \{1, 2, \ldots, r_i - 1\}$ and mark the corresponding vertices of H.
6. (a) Label all vertices of G according to the rule in Fact 3.4 and mark the corresponding vertex of H.
   (b) If a vertex is marked more than once then reject.
7. Scan all edges of H in some order and check whether they correspond to an edge in G.

Correctness of the algorithm follows from the previous discussion. Concerning the time complexity, note first that steps 1-5 of the algorithm can clearly be performed in $O(m)$ time. In particular, in Step 4 (a) we only need to count the number of edges in the C_i's. The labelling algorithm, which is by Proposition 3.6 of complexity $O(m)$, is the essential part of Step 6. The rest can be done in $O(m)$ time because we just need to point from an already labelled vertex of G to a corresponding vertex of H. Thus, the following theorem holds.

Theorem 3.7. *For a given graph G on n vertices and m edges one can decide in $O(m)$ time and $O(n^2)$ space whether G is a Hamming graph. The time complexity is optimal.*

3.2 Quasi-median graphs

Median graphs were introduced as graphs in which every triple of vertices has a unique median. Mulder [28] introduced quasi-median graphs as a generalization of median graphs in the following way.

Let (u_1, u_2, u_3) be a triple of vertices of a graph G. A *quasi-median* of (u_1, u_2, u_3) is a triple (x_1, x_2, x_3) such that for any distinct i and j

(i) $d(u_i, u_j) = d(u_i, x_i) + d(x_i, x_j) + d(x_j, u_j)$,
(ii) $d(x_i, x_j) = k$,

where k is minimal with respect to (i) and (ii). G is a *quasi-median* graph if it satisfies the following conditions:

(i) any triple of vertices in G has a unique quasi-median,
(ii) G does not contain $K_4 - e$ as an induced subgraph,
(iii) the convex hull of any isometric C_6 is Q_3.

Note that if $k = 0$ the quasi-median reduces to a median of a considered triple of vertices.

Median graphs were characterized in Theorem 2.5 as (weak) retracts of binary Hamming graphs. That the definition of quasi-median graphs due to Mulder is really the most natural generalization of median graphs is supported by the following theorem. It was proved independently by Chung, Graham and Saks [12] and Wilkeit [32].

Theorem 3.8. [12, 32] *A graph G is a quasi-median graph if and only if G is a weak retract of a Hamming graph.*

For several other characterizations of quasi-median graphs we refer to [8].

Mulder [28], Chung, Graham and Saks [12] as well as Wilkeit [32] observed that these characterizations lead to polynomial recognition algorithms for this class of graphs. But for a more efficient algorithm an insight due to Hagauer [21] was helpful.

For a graph G and a vertex $s \in V(G)$ let the *skeleton* G_s of G (with respect to s) be the graph we obtain from G by removing all edges uv for which $d(s, u) = d(s, v)$. Note that if G is connected so is G_s. The following result is the principal observation for a fast algorithm for recognizing quasi-median graphs.

Theorem 3.9. [21] *A skeleton of a quasi-median graph is a median graph.*

The recognition algorithm for quasi-median graphs then proceeds as follows. For a given graph G and an arbitrary vertex s of G we first check if G_s is a median graph. For this we can use any algorithm for recognizing median graphs and by Theorem 2.5 this can be done in $O(n^{\frac{3}{2}} \log n)$ time. Furthermore, there exists a binary Hamming labelling γ of G. We can find it in $O(m \log n) = O(n \log^2 n)$ time using the approach from [22].

When we know that G_s is a median graph and that we have a binary labelling γ, we must verify whether the remaining edges fit into the skeleton. To explain this in more detail we need some definitions.

A clique Q of G is *s-gated*, if there exists a vertex x of Q such that $d(s, x) = d(s, y) - 1$ for any vertex y of Q, $y \neq x$. We then define a relation S on $E(G)$ as follows. Edges e and f are in relation S if they belong to the same s-gated triangle. Let S^* be the transitive closure of S. We now introduce another relation T defined on $E(G_s)$. Edges e and f of $E(G_s)$ are in relation T, if there is an edge g of $E(G_s)$ such that eSg and the γ labels of endvertices of g and f differ in the same coordinate. With these two relations we can characterize quasi-median graphs as follows.

Theorem 3.10. **[21]** *Let G be a connected graph and $s \in V(G)$. Then G is a quasi-median graph if and only if the following conditions hold:*
(i) G_s is a median graph,
(ii) each equivalence class of S^* induce an s-gated clique,
(iii) T is an equivalence relation.

By a result from [24] it can be shown that (ii) and (iii) can be checked in $O(m \log n)$ time. So we have:

Theorem 3.11. **[21]** *Let $MG(n)$ denote the complexity of recognizing median graphs on n vertices. Then, for a given graph G on n vertices and m edges, one can decide in $O(MG(n) + m \log n)$ steps whether G is a quasi-median graph.*

3.3 The Graham and Winkler embedding

Before we consider the last class of graphs, partial Hamming graphs, we briefly describe the canonical embedding of a graph into a Cartesian product due to Graham and Winkler [20]. For a more detailed treatment and proofs we refer to the original paper of Graham and Winkler [20] and to [19, 23, 34] for related results.

Let $E_1, E_2, \ldots, E_k$ be the equivalence classes of the relation Θ^*. For $i = 1, 2, \ldots, k$ let G_i denote the graph $(V(G), E(G)\backslash E_i)$ and let $C_{i,1}, C_{i,2}, \ldots, C_{i,m_i}$ denote the connected components of G_i. Form the graphs G_i^*, $i = 1, 2, \ldots, k$, by setting $V(G_i^*) = \{C_{i,1}, C_{i,2}, \ldots, C_{i,m_i}\}$ and by taking $C_{i,j}C_{i,j'}$ to be an edge of G_i^* if some edge in E_i joins a vertex in $C_{i,j}$ to a vertex in $C_{i,j'}$.

We now define a natural contraction $\alpha_i : V(G) \to V(G_i^*)$ by setting $\alpha_i(v) = C_{i,j}$ if $v \in C_{i,j}$. We thus obtain a mapping

$$\alpha : V(G) \to \prod_{i=1}^{k} G_i^*,$$

where

$$\alpha(v) = (\alpha_1(v), \alpha_2(v), \ldots, \alpha_k(v)).$$

The mapping α is the *canonical embedding* of a graph into a Cartesian product of graphs. Its most important property is:

Theorem 3.12. [20] *The canonical embedding is an isometric embedding of G into the Cartesian product $\prod_{i=1}^{k} G_i^*$.*

The embedding α possesses several other properties which are collected in Theorem 3.13.

We call an isometric embedding $\beta : G \to \prod_{i=1}^{m} H_i$ *irredundant* if $|H_i| \geq 2$ holds for $i = 1, 2, \ldots, m$, and if the vertex h occurs as a coordinate value of the image of some

$g \in V(G)$ for all $h \in V(H_i)$. This means that there are no unused factors or vertices in an irredundant embedding.

Furthermore, let us call a graph G *irreducible* if for any irredundant isometric embedding $\beta : G \to \prod_{i=1}^{m} H_i$ $m = 1$ and $G = H_1$.

Theorem 3.13. *Let α be the canonical embedding of a connected graph G. Then*

(i) α is irredundant,

(ii) α has the largest possible number of factors among all irredundant isometric embeddings of G,

(iii) each factor G_i^* is irreducible,

(iv) α is unique among the embeddings from (ii).

In the next section we show how α can be used to obtain a simple recognition algorithm for partial Hamming graphs.

3.4 Partial Hamming graphs

Graphs that can be isometrically embedded into a Hamming graph are called *partial Hamming graphs*. Alternatively, G is a partial Hamming graph if each vertex of G can be labelled by a word of fixed length over some alphabet such that the distance between any two vertices of G is equal to the Hamming distance between the corresponding words. Quasi-median graphs are partial Hamming graphs. Furthermore, the graph which is obtained from the Cartesian product of K_2 by K_3 by removing a vertex is a partial Hamming graph but not a quasi-median graph. Thus, partial Hamming graphs form a proper extension of quasi-median graphs.

In [33] Winkler proved that any two isometric embeddings of a graph into a Hamming graph are equivalent (in a technical sense). This result also yielded a simple $O(n^5)$ recognition algorithm for recognizing partial Hamming graphs. Later, Wilkeit [31] obtained several characterizations of partial Hamming graphs and an $O(n^3)$ recognition algorithm. In addition we recall that partial Hamming graphs were also characterized by Chepoi in [10].

Winkler's algorithm was recently modified by Aurenhammer, Formann, Idury, Schäffer and Wagner [1] to run in $O(D(m, n) + n^2)$ time, where $D(m, n)$ denotes the time needed to compute the distance matrix of a graph. Thus, in general the complexity is $O(mn)$. Here we will describe another $O(mn)$ algorithm due to Imrich and Klavžar [24] which is very simple to formulate but we need some background to explain the idea.

As indicated in Section 2.3 we shall now explain how to compute Θ^* efficiently by a method of Feder [17]. Let T be a spanning tree of a graph G. We say the edges $e, e' \in E(G)$ are in relation Θ_1 if they are in relation Θ and if at least one of the edges e, e' belongs to T. Most importantly, Feder showed that $\Theta^* = \Theta_1^*$. Thus, instead of computing Θ^* it is enough to compute Θ_1^*. This can be done in $O(mn)$ time since we can calculate the distances from a vertex to all other vertices in $O(m)$ time.

Using the above mentioned result of Winkler from [33] the following crucial theorem for the algorithm was proved in [24].

Theorem 3.14. *Let $\beta : G \to \prod_{i=1}^{m} H_i$ be an isometric irredundant embedding of a graph G into a product of complete graphs H_i. Then this embedding is the canonical isometric embedding.*

Thus, for a given connected graph G we compute Θ_1^* and the graphs G_i, $i = 1, 2, \ldots, k$. Then G is a partial Hamming graphs if and only if all the G_i are complete graphs. In addition, if G is a partial Hamming graph, then we can obtain a corresponding labelling from α. So:

Theorem 3.15. **[1, 24]** *For a given graph G on n vertices and m edges one can decide in $O(nm)$ steps whether G is a partial Hamming graph.*

Main algorithmic results of this section are summarized in Table 2. Recall that $MG(n)$ denotes the complexity of recognizing median graphs on n vertices.

class of graphs	time complexity
Hamming graphs	m
quasi-median graphs	$m \log n + MG(n)$
partial Hamming graphs	mn

Table 2: Complexities for the general case

4 Concluding remark

In this paper we have considered recognition algorithms pertaining to graphs arising in the following hierarchy:

Hamming graphs ⇒ quasi-median graphs ⇒ partial Hamming graphs

Where we stopped in the binary case, another hierarchy begins, the so-called ℓ^1-hierarchy, cf. [5, 14] and references there. It starts (for graphs) with partial binary Hamming graphs and stops with graphs with one positive eigenvalue of their distance matrix. More precisely (cf. [5]) it contains the following classes of graphs:

graphs embeddable in a hypercube ⇒ graphs embeddable in ℓ_1 ⇒ hypermetric graphs ⇒ graphs of negative type ⇒ graphs with one positive eigenvalue

Although the hierarchy is strict, it collapses for bipartite graphs to the one considered in Section 2 as proved by Roth and Winkler [29]. More precisely, they proved that a graph G is a partial binary Hamming graph if and only if G is bipartite and has one positive eigenvalue. In contrast to the hierarchy considered in this paper the ℓ^1-hierarchy is mostly unexplored with respect to efficient recognition algorithms. It should be noted, however, that Shpectorov [30] proved that there is a polynomial algorithm for recognizing l_1-graphs and that at the "Discrete Metric Spaces" conference in Bielefeld (November 1994) he announced the complexity $O(nm)$. This result has been recently documented in [15].

References

[1] F. Aurenhammer, M. Formann, R. Idury, A. Schäffer and F. Wagner, Faster isometric embeddings in products of complete graphs, *Discrete Appl. Math.* **52** (1994), 17–28.

[2] F. Aurenhammer and J. Hagauer, Recognizing binary Hamming graphs in $O(n^2 \log n)$ time, in: Proc. 16th Int. Workshop on Graph Theoretical Concepts in Computer Science, *Lecture Notes in Comput. Sci.* **484** (Springer, New York, 1991), 90–98. Also: Math. Systems Theory, to appear.

[3] F. Aurenhammer and J. Hagauer, Computing equivalence classes among the edges of a graph with applications, *Discrete Math.* **109** (1992), 3–12.

[4] F. Aurenhammer, J. Hagauer and W. Imrich, Cartesian graph factorization at logarithmic cost per edge, *Comput. Complexity* **2** (1992), 331–349.

[5] D. Avis and H. Maehara, Metric extensions and the L_1 hierarchy, *Discrete Math.* **131** (1994), 17–28.

[6] H.-J. Bandelt, Retracts of hypercubes, *J. Graph Theory* **8** (1984), 501–510.

[7] H.-J. Bandelt, Characterization of Hamming graphs, manuscript, August 1992.

[8] H.J. Bandelt, H.M. Mulder in E. Wilkeit, Quasi-median graphs and algebras, *J. Graph Theory* **18** (1994), 681–703.

[9] K.V.S. Bhat, On the complexity of testing a graph for n-cube, *Inform. Process. Lett.* **11** (1980), 16–19.

[10] V. D. Chepoi, d-convexity and isometric subgraphs of Hamming graphs, *Cybernetics* **1** (1988), 6–9.

[11] F.R.K. Chung, R.L. Graham, and M.E. Saks, Dynamic search in graphs, in: *Discrete Algorithms and Complexity* (H. Wilf, ed.) (Academic Press, New York, 1987), 351–387.

[12] F.R.K. Chung, R.L. Graham, and M.E. Saks, A dynamic location problem for graphs, *Combinatorica* **9** (1989), 111–132.

[13] M. Deza and V.P. Grishukhin, Hypermetric graphs, *Quart. J. Math. Oxford Ser. (2)* **44** (1993), 399–433.

[14] M. Deza and M. Laurent, ℓ_1-rigid graphs, *J. Algebraic Combin.* **3** (1994), 153–175.

[15] M. Deza and S. Shpectorov, Recognition of the ℓ_1-graphs with complexity $O(nm)$, or football in a hypercube, *Rapport de Recherche du LIENS*, May (1995).

[16] D. Djoković, Distance preserving subgraphs of hypercubes, *J. Combin. Theory Ser. B* **14** (1973), 263–267.

[17] T. Feder, Product graph representations, *J. Graph Theory* **16** (1992), 467–488.

[18] M. R. Garey and R. L. Graham, On cubical graphs, *J. Combin. Theory Ser. B* **18** (1975), 84–95.

[19] R. L. Graham, Isometric embeddings of graphs, in: *Selected Topics in Graph Theory* III, (L.W. Beineke and R.J. Wilson, eds.) (Academic Press, London, 1988), 133–150.

[20] R. L. Graham and P. M. Winkler, On isometric embeddings of graphs, *Trans. Amer. Math. Soc.* **288** (1985), 527–536.

[21] J. Hagauer, Skeletons, recognition algorithm and distance matrix of quasi-median graphs, *Intern. J. Computer Math.* **55** (1995), 155–171.

[22] J. Hagauer, W. Imrich and S. Klavžar, Recognizing graphs of windex 2, Preprint Series Univ. Ljubljana **31** (1993), 410.

[23] W. Imrich, Embedding graphs into Cartesian products, *Ann. New York Acad. Sci.* **576** (1989), 266–274.

[24] W. Imrich and S. Klavžar, A simple $O(mn)$ algorithm for recognizing Hamming graphs, *Bull. Inst. Comb. Appl.* **9** (1993), 45–56.

[25] P.K. Jha and G. Slutzki, An $O(n^2 \log n)$ algorithm for recognizing median graphs, TR#89-4, Iowa State University, May 1989.

[26] P.K. Jha and G. Slutzki, Convex-expansion algorithms for recognition and isometric embedding of median graphs, *Ars Combin.* **34** (1992), 75–92.

[27] H.M. Mulder, The structure of median graphs, *Discrete Math.* **24** (1978), 197–204.

[28] H.M. Mulder, The Interval Function of a Graph, Mathematical Centre Tracts 132, Mathematisch Centrum, Amsterdam, 1980.

[29] R.L. Roth and P.M. Winkler, Collapse of the metric hierarchy for bipartite graphs, *European J. Combin.* **7** (1986), 371–375.

[30] S.V. Shpectorov, On scale embeddings of graphs into hypercubes, *European J. Combin. 14* (1993), 117–130.

[31] E. Wilkeit, Isometric embeddings in Hamming graphs, *J. Combin. Theory Ser. B* **50** (1990), 179–197.

[32] E. Wilkeit, The retracts of Hamming graphs, *Discrete Math.* **102** (1992), 197–218.

[33] P. Winkler, Isometric embeddings in products of complete graphs, *Discrete Appl. Math.* **7** (1984), 221–225.

[34] P. Winkler, The metric structure of a graph: Theory and applications, in: Eleventh British Combinatorial Conference, Surveys in Combinatorics 1987 (C. Whitehead, ed.), *London Math. Soc. Lecture Note Ser.* **123** (Cambridge University Press, Cambridge, 1987), 197–221.