

Recognizing Hamming graphs in $O(m)$ time and $O(n^2)$ space

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

Cartesian products of complete graphs are also known as Hamming graphs. Given a graph G on n vertices and m edges we propose an algorithm that decides in $O(m)$ time and $O(n^2)$ space whether G is a Hamming graph.

1 Introduction

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

We begin with the definitions of the justmentioned classes of graphs and then list fast algorithms for recognizing them. We note that the terminology is not uniform and that our definitions are chosen for brevity.

*This work was supported in part by the Ministry of Science and Technology of Slovenia under the grant P1-0206-101.

Thus, *Hamming graphs* can be defined as Cartesian products of complete graphs, *quasimedial graphs* as weak retracts of Hamming graphs and *partial Hamming graphs* as isometric subgraphs of Hamming graphs.

Every Hamming graph is a quasimedial graph and every quasimedial graph a partial Hamming graph.

In case we restrict attention to bipartite graphs in the above classes one obtains, first of all, *binary Hamming graphs*, which are just bipartite Hamming graphs or Cartesian products of K_2 -s. These graphs are also known as *hypercubes*.

Median graphs are then defined as retracts of binary Hamming graphs and can also be characterized as bipartite quasimedial graphs.

Finally, *partial binary Hamming graphs* are isometric subgraphs of binary Hamming graphs.

As expected, every binary Hamming graph is a median graph and every median graph a partial binary Hamming graph.

Let G be a connected graph on n vertices with m edges. Then one can determine whether G is a binary Hamming graph in $O(m)$ time as has been shown by Bhat [3].

For median graphs the situation is less favorable, but they can be recognized in $O(mn^{1/2})$ time as has been shown by Hagauer, Imrich and Klavžar [4].

For the recognition of partial binary Hamming graphs several algorithms have been proposed, all with complexity $O(mn)$. These algorithms include one by Jha and Slutzky [6], one by Bandelt [] and one by Imrich and Klavžar [5].

Since the number of edges of a binary Hamming graph with n vertices is bounded by $n \log n$ these complexities actually are $O(n \log n)$ for the recognition of binary Hamming graphs, $O(n^{3/2} \log n)$ for median graphs and $O(n^2 \log n)$ for partial binary Hamming graphs.

For Hamming graphs the fastest algorithm hitherto uses a prime factorization algorithm of connected graphs with respect to the Cartesian product [2] and has complexity $O(m \log n)$. For quasimedial graphs there is an algorithm by Hagauer [] of complexity $O(m \log n + n^{3/2} \log n)$ and for partial Hamming graphs there are several algorithms of complexity $O(mn)$, see e.g. [5].

The purpose of this note is an algorithm that checks in $O(m)$ time and $O(n^2)$ space whether a graph is a Hamming graph.

2 Hamming graphs

To fix notation we begin with a more detailed definition of Hamming graphs.

Let r_1, \dots, r_t be given integers ≥ 2 and let V be the set of t -tuples $a_1a_2\dots a_t$ with $0 \leq a_i \leq r_i - 1$. 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 $a_1a_2\dots a_t$ and $b_1b_2\dots b_t$ 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 the graph $H = (V, E)$ is called a Hamming graph.

It is easy to see that the shortest path distance in H between any two vertices $a_1a_2\dots a_t$ and $b_1b_2\dots b_t$ is the number of places (or components) in which these t -tuples differ. This distance is also called the *Hamming distance* and the corresponding labelling of the vertices of H a *Hamming labelling*.

Let $v_0 = 00\dots 0$ and let v_0, v_1, \dots, 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. It consists 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 $a_1a_2\dots a_t$ in which exactly k of the a_i are $\neq 0$. For further reference we state the following observations as Facts.

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

$$h : v \mapsto a_1a_2\dots a_i\dots a_t$$

is a Hamming labelling of H , then

$$\pi h : v \mapsto a_1a_2\dots \pi a_i\dots a_t$$

is also a Hamming labelling.

Fact 2 *Let $1 \leq i < j \leq t$ and h be given as in Fact 1. Then*

$$h_{ij} : v \mapsto a_1a_2\dots a_{i-1}a_ja_{i+1}\dots a_{j-1}a_ia_{j+1}\dots a_t$$

is also a Hamming labelling.

Fact 3 *The vertices of type $0\dots 0a_i0\dots 0, a_i \neq 0$, 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 4 *Let $u = a_1a_2\dots a_t \in l_k, k \geq 1$. 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 $v = b_1b_2\dots b_t$ and $w = c_1c_2\dots c_t$ then $a_i = \max\{b_i, c_i\}$ for $i = 1, \dots, t$.

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.

Output: A Hamming labelling of H .

- (1) Choose a vertex v_0 .
- (2) Arrange the vertices of H in levels l_0, l_1, \dots, 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, \dots, C_t with $r_1 - 1, r_2 - 2, \dots, r_t - 1$ 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 $0\dots 0a_i0\dots 0$, i.e. vectors of length t in which only the i -th coordinate a_i is different from zero, such that 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 $b_1b_2\dots b_t$ and $c_1c_2\dots c_t$, respectively. Setting $a_i = \max\{b_i, c_i\}$ we obtain a label $a_1a_2\dots a_t$ for u .

Proposition 5 *The labelling algorithm, applied to a Hamming graph H , yields a Hamming labelling of H .*

Proof. By Fact 1 there is a Hamming labelling where v_0 has the label $00\dots 0$.

By Fact 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 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 4. \square

Proposition 6 *The time complexity of the labelling algorithm is $O(m)$, where $m = |E(H)|$.*

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

Steps (2) and (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, $O(nt) = O(m)$. \square

Thus, for Hamming graphs H our 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. How and how fast can we check this?

We may assume that all labels of the form $a_1a_2\dots a_t$ with $a_i \in \{0, 1, \dots, r_i - 1\}$ 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, \dots, 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, \dots, C_t with $r_1-1, r_2-2, \dots, r_t-1$ vertices, respectively.

(4.a) If any of the subgraphs of G spanned by the C_i is not complete then reject.

(4.b) If $n \neq \prod_{i=1}^t r_i$ then reject.

(4.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.

(4.d) Form the vertices of the Hamming graph H with the labels $a_1 a_2 \dots a_t$ where $a_i \in \{0, \dots, r_i - 1\}$.

(4.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 $0 \dots 0 a_i 0 \dots 0$, where $a_i \in \{1, \dots, r_i - 1\}$ and mark the corresponding vertices of H .

(6.a) Label all vertices of G according to the rule in Fact 4 and mark the corresponding vertex of H .

(6.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 .

References

- [1] F. Aurenhammer, M. Formann, R. Idury, A. Schäffer and F. Wagner, Faster isometric embeddings in products of complete graphs, *Discrete Appl. Math.*, to appear.
- [2] F. Aurenhammer, J. Hagauer and W. Imrich, Cartesian graph factorization at logarithmic cost per edge, *Comput. Complexity* **2** (1992) 331–349.
- [3] K.V.S. Bhat, On the complexity of testing a graph for n -cube, *Inform. Process. Lett.* **11** (1980) 16–19.
- [4] J. Hagauer, W. Imrich and S. Klavžar, Recognizing graphs of windex 2, Preprint Series Univ. Ljubljana **31** (1993) 410.
- [5] W. Imrich and S. Klavžar, A simple $O(mn)$ algorithm for recognizing Hamming graphs, *Bull. Inst. Comb. Appl.* **9** (1993), 45–56.
- [6] P.K. Jha and G. Slutzki, Convex-expansion algorithms for recognition and isometric embedding of median graphs, *Ars Combin.* **34** (1992), 75–92.