

# Recognizing median graphs in subquadratic time

Johann Hagauer  
Technical University Graz  
IGI, Klosterwiesgasse 32 /II  
8010 Graz, Austria

Wilfried Imrich  
Institut für Mathematik und  
Angewandte Geometrie  
Montanuniversität Leoben  
8700 Leoben, Austria

Sandi Klavžar \*  
Department of Mathematics, PEF  
University of Maribor  
Koroška cesta 160  
2000 Maribor, Slovenia

## Abstract

Motivated by a dynamic location problem for graphs, Chung, Graham and Saks introduced a graph parameter called windex. Graphs of windex 2 turned out to be, in graph-theoretic language, retracts of hypercubes. These graphs are also known as median graphs and can be characterized as partial binary Hamming graphs satisfying a convexity condition. In this paper an  $O(n^{\frac{3}{2}} \log n)$  algorithm is presented to recognize these graphs. As a by-product we are also able to isometrically embed median graphs in hypercubes in  $O(m \log n)$  time.

## 1 Introduction

To any three vertices  $u, v, w$  in a tree there exists a vertex  $x$  which lies on shortest paths between any two of them. Such a vertex is called a median of  $u, v$  and  $w$ . It is easily seen that medians are unique in trees. In general, we call a graph a median graph if any triple of vertices has a unique median. Trees and hypercubes are median graphs, but not nonbipartite graphs. In this paper we focus on a fast recognition algorithm for this class of graphs. Of course, our methods are based on previous results.

Pioneering work on median graphs was done by Avann [1] and Nebeský [22]. More extensive investigations (see the reference list) of these graphs followed by Mulder and Bandelt. In fact, Mulder independently introduced the notion of median graphs. He and his co-workers obtained many interesting results on this class of graphs, see [6, 18, 19, 20, 21]. Mulder showed among other results that median graphs are precisely those graphs that can be obtained from a one-vertex graph by a so-called convex expansion procedure [18, 19]. Mulder also observed

---

\*This work was supported in part by the Ministry of Science and Technology of Slovenia under the grants P1-0206-101 and J1-7036.

that median graphs are closely related to distributive lattices and graph retracts; see Avann [1], Bandelt [4], and Duffus and Rival [9].

Median graphs are of considerable relevance in optimization theory. For recent applications we refer to McMorris, Mulder and Roberts, [17] and to references there.

Motivated by a dynamic location problem for graphs, Chung, Graham and Saks [7, 8] introduced a graph parameter called *windex*. In their context median graphs appear as graphs of windex 2. It is natural to ask for an efficient algorithm for the recognition of median graphs. As a by-product of their investigation, Chung, Graham and Saks [7] proposed an  $O(n^4)$  algorithm. Jha and Slutzki [15, 16] followed with two different approaches, each yielding an  $O(n^2 \log n)$  algorithm. In [15] they adapted Bandelt's approach [4] using retracts and in [16] Mulder's convex expansion method. A simple algorithm of the same complexity was recently proposed by Imrich and Klavžar [14].

Median graphs can be characterized as retracts of hypercubes (see Bandelt [4]). Thus, they are isometric subgraphs of hypercubes. Isometric subgraphs of hypercubes are also known as partial binary Hamming graphs and algorithms for recognizing them are of interest for the problem of recognizing median graphs. Also, for all these graphs the number  $m$  of edges is at most  $n \log n$ . This implies  $O(mn) = O(n^2 \log n)$ , which is the complexity of the abovementioned algorithm of Jha and Slutzki. The  $O(mn)$  bound is also achieved by algorithms for recognizing partial (binary) Hamming graphs, see [2, 3, 10, 12] for particular results and [13] for a recent survey. So far no algorithm with subquadratic time bound is known for recognizing partial (binary) Hamming graphs or median graphs. The main contribution of this paper is an  $O(n^{\frac{3}{2}} \log n)$  algorithm for recognizing median graphs (alias graphs of windex 2).<sup>1</sup> The method used also yields an algorithm which isometrically embeds a given median graph into a hypercube in  $O(m \log n)$  time.

The paper is organized as follows: In Section 2 all results needed for the algorithm are collected or proved. In Section 3 the algorithm is presented and its correctness shown. The time complexity is established in Section 4.

## 2 Preliminaries

All graphs considered in this paper are finite undirected graphs without loops and multiple edges. Throughout the paper, for a given graph  $G$ , let  $n$  and  $m$  stand for the number of its vertices and edges, respectively. For  $u \in V(G)$  let  $N(u)$  be the set of all vertices adjacent to  $u$  and let  $d(u) = |N(u)|$  be the *degree* of  $u$ . For  $X \subseteq V(G)$ ,  $\langle X \rangle$  denotes the subgraph induced by  $X$ . For  $u, v \in V(G)$ ,  $d_G(u, v)$  or  $d(u, v)$ , if  $G$  is understood, denotes the length of a shortest path in  $G$  from  $u$  to  $v$ . For  $X \subseteq V(G)$  and  $v \in V(G)$ ,  $d(v, X)$  denotes the distance from the vertex  $v$  to  $X$  and is defined by  $d(v, X) = \min_{u \in X} d(v, u)$ .

Let  $G$  be a graph and let  $u, v \in V(G)$ . Then

$$I(u, v) = \{w \in V(G) \mid w \text{ lies on a shortest } u - v \text{ path}\}$$

is called an *interval* of  $G$  between  $u$  and  $v$ . A subgraph  $H$  of  $G$  is *convex*, if for any  $u, v \in V(H)$ ,  $I(u, v) \subseteq V(H)$ . A *median* of a set of three vertices  $u, v$  and  $w$  is a vertex that lies in

<sup>1</sup>Recently the authors discovered connections with the problem of recognizing triangle-free graphs which make it plausible that the bound is optimal.

$I(u, v) \cap I(u, w) \cap I(v, w)$ . Alternatively,  $x$  is a median of  $u$ ,  $v$  and  $w$  if

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

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

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

A connected graph  $G$  is a *median graph* if every triple of its vertices has a unique median.

As we already mentioned in the introduction, trees and hypercubes are median graphs. Hypercubes are also known as  $n$ -cubes. The vertex set of an  $n$ -cube consists of all binary words of length  $n$ , two such words being adjacent if they differ in exactly one place. Thus, 001 and 011 are adjacent in the 3-cube but not 001 and 100.

Let  $G$  be a connected graph and let  $ab \in E(G)$ . The following sets will play an important role:

$$\begin{aligned} W_a &:= \{w \in V \mid d(w, a) < d(w, b)\}, \\ W_b &:= \{w \in V \mid d(w, b) < d(w, a)\}, \\ U_a &:= \{u \in W_a \mid u \text{ is adjacent to a vertex in } W_b\}, \\ U_b &:= \{u \in W_b \mid u \text{ is adjacent to a vertex in } W_a\}, \\ F &:= [U_a, U_b] = \{uv \mid u \in U_a, v \in U_b\}. \end{aligned}$$

Sometimes we shall also write  $F_{ab}$  to indicate the set  $F$ . Note that  $V = W_a \cup W_b$  for bipartite graphs.

A subgraph  $H$  of a graph  $G$  is an *isometric* subgraph, if  $d_H(u, v) = d_G(u, v)$  for all  $u, v \in V(H)$ . If  $H'$  is a graph which is isomorphic to an isometric subgraph  $H$  of  $G$  we say that  $H'$  can be *isometrically embedded* into  $G$ .

In the next theorem we summarize important structural properties of median graphs.

**Theorem 2.1** [18, 19, 22] *Let  $G$  be a median graph and let  $ab \in E(G)$ . Then*

- (i)  $G$  is bipartite and contains no  $K_{2,3}$  as an induced subgraph;
- (ii)  $G$  can be isometrically embedded in a hypercube;
- (iii)  $\langle U_a \rangle$ ,  $\langle U_b \rangle$ ,  $\langle W_a \rangle$  and  $\langle W_b \rangle$  are convex subgraphs of  $G$ ;
- (iv)  $F$  is a matching on  $U_a, U_b$  that defines an isomorphism between  $\langle U_a \rangle$  and  $\langle U_b \rangle$ .

The next theorem is the basis for our algorithm.

**Theorem 2.2** *Let  $G$  be a connected bipartite graph, and let  $ab \in E(G)$ . Suppose the following properties hold:*

- (i)  $F$  is a matching that defines an isomorphism between  $\langle U_a \rangle$  and  $\langle U_b \rangle$ ;
- (ii) for any  $u \in U_a$  and  $v \in U_b$ ,  $I(u, a) \subseteq U_a$  and  $I(v, b) \subseteq U_b$ , respectively;
- (iii) for any  $u \in W_a \setminus U_a$  and  $v \in W_b \setminus U_b$ ,  $|N(u) \cap U_a| \leq 1$  and  $|N(v) \cap U_b| \leq 1$ .

*Then  $G$  is a median graph if and only if  $\langle W_a \rangle$  and  $\langle W_b \rangle$  are median graphs.*

Before giving the proof we first show three lemmas. The first one is due to Bandelt (personal communication to Jha and Slutzki in [16]). 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 2.3** *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, it is possible to replace “isometric” by “connected” in the formulation of Lemma 2.3, cf. [13].

**Lemma 2.4** *Let  $G$  be a median graph and let  $H$  be a convex subgraph of  $G$ . Then for any  $v \in G$  there exists a unique  $u \in H$  such that  $d(v, H) = d(v, u)$ .*

**Proof.** Clearly the statement holds for  $v \in H$ . Assume  $v \in G$  but  $v \notin H$  and that  $d(v, H) = d(v, u) = d(v, w)$  for some  $u, w \in H$ ,  $u \neq w$ . Let  $y$  be the median in  $G$  of the triple  $v, u, w$ . By our assumption this median exists and is unique. Since  $H$  is convex,  $y \in H$ . But then  $w \neq u$  implies either  $d(v, H) \leq d(v, y) < d(v, u)$  or  $d(v, H) \leq d(v, y) < d(v, w)$ , a contradiction.  $\square$

**Lemma 2.5** *Let  $G$  be a median graph and let  $H$  be a convex subgraph of  $G$ . For any  $v \in G$  and for any  $w \in H$ ,  $d(v, w) = d(v, u) + d(u, w)$ , where  $u \in H$  is the unique vertex satisfying  $d(v, H) = d(v, u)$ .*

**Proof.** If  $u = v$  or  $u = w$  the lemma holds. For  $u \neq v$  and  $u \neq w$ , note that  $v \neq w$ , and let  $y$  be the median of the triple  $u, v, w$ . Hence  $d(v, u) = d(v, y) + d(y, u)$ . By convexity,  $y \in H$  and therefore either  $y = u$  or  $d(v, y) < d(v, u)$ , a contradiction. Thus we have  $y = u$  and  $d(v, w) = d(v, u) + d(u, w)$ .  $\square$

**Proof of Theorem 2.2.** If  $G$  is a median graph, then by Theorem 2.1 (iii)  $W_a$  and  $W_b$  are convex subgraphs of  $G$ . It is easy to see that a convex subgraph of a median graph is itself a median graph.

For the converse note first that by Lemma 2.3, (ii) and (iii) imply that  $\langle U_a \rangle$  and  $\langle U_b \rangle$  are convex in  $W_a$  and  $W_b$  respectively. But it is also easy to see this directly. Therefore Lemmas 2.4 and 2.5 can be applied. We must show that for any triple  $v_1, v_2$  and  $v$  there is a unique median in  $G$ . If  $v_1, v_2$  and  $v$  all belong to  $W_a$  or to  $W_b$  little has to be proved: only that no vertex of  $W_b$  can be a median for three vertices of  $W_a$ , and similarly for triples of vertices in  $W_b$ . Hence, we may assume without loss of generality that  $v_1, v_2 \in W_b$  and  $v \in W_a$ . Let  $u$  be the vertex in  $U_a$  such that  $d(v, U_a) = d(v, u)$ , let  $uu'$  be the matching edge, where  $u' \in U_b$ , and let  $y$  be the unique median of the triple  $v_1, v_2, u'$  in  $W_b$ . By the fact that any shortest path from  $v_1$  or  $v_2$  to  $v$  must contain a matching edge and by Lemma 2.5 the following holds:

$$d(v_1, v) = d(v_1, y) + d(y, y') + d(y', u') + d(u', u) + d(u, v)$$

$$d(v_2, v) = d(v_2, y) + d(y, y') + d(y', u') + d(u', u) + d(u, v),$$

where  $y' \in U_b$  satisfies  $d(y, U_b) = d(y, y')$ . (Note that this holds in particular when  $v_1$  or  $v_2$  is equal to  $u'$  since then  $y = y' = u'$ .) Thus,  $y$  is a median for  $v_1, v_2, v$ . To demonstrate that no other vertex  $y^*$  can be a median for  $v_1, v_2, v$ , note that  $y^*$  would have to belong to  $W_b$  since it would lie on a shortest path between  $v_1$  and  $v_2$ ; also note that  $y^*$  would be a median in  $W_b$  for  $v_1, v_2$  and  $u'$ . Since  $W_b$  is a median graph,  $y^* = y$ .  $\square$

### 3 The algorithm

The main idea of our algorithm is to decompose a graph into two parts, check if the conditions of Theorem 2.2 are fulfilled and recursively repeat the procedure until we end up with a single vertex graph. Jha and Slutzki [16] used a similar idea, but instead of decomposing a graph into two parts they contracted it along  $F$ .

To improve the running time of the algorithm we introduce the following sets:

$$\begin{aligned} CU_a &:= \{u \in U_a \mid \text{there is a path in } \langle U_a \rangle \text{ from } a \text{ to } u\}, \\ CU_b &:= \{u \in U_b \mid \text{there is a path in } \langle U_b \rangle \text{ from } b \text{ to } u\}, \\ CW_a &:= \{u \in W_a \setminus U_a \mid \text{there is an edge } ux, x \in CU_a\}, \\ CW_b &:= \{u \in W_b \setminus U_b \mid \text{there is an edge } ux, x \in CU_b\}, \\ CF &:= [CU_a, CU_b] = \{uv \mid u \in CU_a, v \in CU_b\}. \end{aligned}$$

Our algorithm tries to identify these sets without necessarily processing the entire graph. We call the sets identified by the algorithm  $CCU_a$ ,  $CCU_b$ ,  $CCW_a$ ,  $CCW_b$  and  $CCF$ . In the case that  $G$  is a median graph the sets computed by the algorithm will be the same as defined above. The sets  $CCU_b$  and  $CCW_b$  will be dynamically defined by the algorithm. Initially  $CCU_b = \{b\}$  and  $CCW_b = \emptyset$ . If  $G$  is a median graph, then at termination  $CU_a$ ,  $CU_b$  and  $CF$  must equal  $U_a$ ,  $U_b$  and  $F$ , respectively. If any of these equalities does not hold, then  $G$  will be rejected by our algorithm.

In case that  $G$  is not a median graph and  $U_a$  and  $U_b$  are not connected the equalities do not hold. For the case when  $G$  is not a median graph but  $U_a$  and  $U_b$  are connected, see the remark after Lemma 3.3.

If  $G$  is a median graph, then our algorithm will disconnect the graph into two connected components,  $\langle W_a \rangle$  and  $\langle W_b \rangle$ , both of which must be median graphs since they are convex. Then we apply the same procedure to both components separately.

We first describe how to compute the sets  $CU_a$ ,  $CU_b$  and  $CF$  efficiently. Let  $a \in V(G)$ . Call a directed edge  $uv$  an *up-edge* (with respect to  $a$ ) if  $d(u, a) < d(v, a)$  and a *down-edge* otherwise. In the following procedure with the parameter directed edge  $ab$ , up- and down-edges refer to  $a$ . We call the edge  $ab$  a *separation edge*.

It is important to note that our final algorithm will start with a vertex  $v_0$  of largest degree and that all edges of the graph are directed with respect to this vertex  $v_0$ . Since we only have to consider bipartite graphs, every edge  $uv$  is either an up-edge (if  $u$  is closer to  $v_0$  than  $v$ ) or a down-edge. This orientation is achieved by a breadth first search (BFS). The algorithm is structured such, that we do not have to alter these directions when we switch to another reference vertex with respect to which we consider up- and down-edges. Moreover, distances from a new reference vertex (say  $a$ ) are obtained by the equality  $d(a, x) = d(v_0, x) - d(v_0, a)$ .

Thus, the following procedure assumes that  $a$  is a root of  $G$  with respect to which all edges have been oriented as up- and down-edges and that all distances to  $a$  are known. The steps of the procedure are justified later, in particular see Lemma 3.4 for Step 2.2.3.

PROCEDURE FINDSETS ( $ab$ )

1.  $CCU_b := \{b\}$ ;  $CCW_b := \emptyset$ ;  $X := \emptyset$ ;
2. while there are unscanned vertices in  $CCU_b$  do
  - 2.1 let  $v \in CCU_b$  be the next vertex in BFS-order;
  - 2.2 for all up-edges  $vu$  of  $v$  do one of the following
    - 2.2.1 if  $u \in CCU_b$  then do nothing;  
 {  $u$  has been correctly classified before }
    - 2.2.2 if  $u$  has no down-edge but  $uv$  then  $CCW_b := CCW_b \cup \{u\}$ ;  
 { if  $u$  were in  $U_b$ , it would have a down-edge in  $F$  }
    - 2.2.3 if  $u$  has more than one down-edge we choose one, say  $uw$ ,  $w \neq v$ , and fix it.  
 For this down-edge do
      - if  $w \in CCW_b$  then  $CCW_b := CCW_b \cup \{u\}$
      - else  $CCU_b := CCU_b \cup \{u\}$ ;
  - 2.3 mark  $v$  as scanned and put all  $u$  such that  $vu$  is a down-edge into  $X$ ;
3. 3.1  $CCU_a := X \setminus CCU_b$ ;
- 3.2  $CCF := \{uv \mid u \in CCU_a, v \in CCU_b\}$ ;

We wish to remark that  $CU_a$ ,  $CU_b$  etc. are defined combinatorially but  $CCU_a$ ,  $CCU_b$  etc. algorithmically. Thus  $u \in CU_a$  etc. stands for the usual set inclusion but  $u \in CCU_a$  etc. means that  $u$  has been classified as being in  $CCU_a$  etc. by the algorithm.

**Lemma 3.1** *If  $G$  is a median graph, then at the conclusion of procedure FINDSETS( $ab$ ),  $CCU_a = U_a$ ,  $CCU_b = U_b$  and  $CCF = F$ .*

**Proof.** We first claim that  $CCU_b \subseteq U_b$ . Suppose on the contrary that when we scan the up-edges of a vertex  $v$ , a vertex  $u$  has been classified as being in  $CCU_b$  but is not in  $U_b$ . We may assume that  $u$  is the first such wrongly classified vertex. Then this could only happen in Step 2.2.3 of FINDSETS. This means that  $u$  has a down-edge  $uw$ , where  $w \notin CCW_b$ . Since  $u \notin U_b$  we have  $w \in W_b$ . Consider the median  $x$  of the triple  $w, v, b$ . As  $d(v, w) = 2$ ,  $x$  is adjacent to both  $v$  and  $w$ . Moreover, the up-edge  $xw$  has been considered before in the BFS order, hence  $w$  was correctly classified as  $w \in CCW_b$ , a contradiction.

We now list several observations which are true for median graphs. In addition to these fact the correctness of the algorithm is also based on the order in which the vertices  $u$  are selected by the algorithm in Step 2.2.

Let  $G$  be a median graph. Then we have:

*Observation 1:* If  $uv_1$  and  $uv_2$  are down-edges and  $v_1, v_2 \in U_b$  then, by convexity of  $U_b$ ,  $u$  is in  $U_b$ .

*Observation 2:* If  $uv_1$  and  $uv_2$  are down-edges and  $v_1 \in U_b$  and  $v_2 \notin CW_b$  then again we have  $u \in U_b$ . This is true because either  $v_2 \in U_b$  or  $v_2 \in U_a$ , in both cases only  $u \in U_b$  possesses this property.

*Observation 3:* If  $uv_1$  and  $uv_2$  are down-edges and  $v_1 \in U_b$  and  $v_2 \in CW_b$  then  $u \in CW_b$ . Clearly,  $u$  is either in  $U_b$  or in  $CW_b$ . However,  $u \in U_b$  would violate the convexity of  $U_b$  as  $v_2$  lies on a shortest path from  $b$  to  $u$ .

*Observation 4:* If  $uv$  is the only down-edge of  $u$  and  $v \in U_b$  then  $u$  is in  $CW_b$ , because in median graphs only vertices in  $CW_b$  have this property.

These observations exhaust the possible cases for down-edges of  $u$ , where  $vu$  is an up-edge of some  $v \in U_b$ . The first three are accounted for in Step 2.2.3, the fourth in 2.2.2.

For the rest of the proof, the order in which the vertices  $u$  are chosen in Step 2.2 is important. Let  $u_1, \dots, u_k$  be the order in which the vertices in Step 2.2 are selected. Let  $u_i$  be the first vertex that is not correctly classified. Obviously,  $u_i \neq b$  and  $u_i \in CW_b$  or  $u_i \in U_b$ , because  $v$  has been correctly classified as being in  $U_b$ . Two cases may occur.

*Case 1:*  $u_i \in U_b$  and  $u_i \in CCW_b$ .

If  $u_i$  has only one down-edge (Step 2.2.2) then  $G$  is not a median graph, because in a median graph all vertices in  $U_b$  except  $b$  have at least two down-edges. If  $u_i$  has a down-edge  $uw$  and  $w \in CCW_b$  and if this edge was chosen in Step 2.2.3 leading to the classification  $u_i \in CCW_b$ , then  $w \in CW_b$ , because  $w$  is classified before  $u$  and hence is correctly classified. But then  $u_i \in U_b$  implies that  $u_i$  has a down-edge  $u_iw$  where  $w \in CW_b$ , which is impossible for a median graph.

*Case 2:*  $u_i \in CW_b$  and  $u_i \in CCU_b$ .

In this case  $u_i$  is classified in Step 2.2.3 and  $u_i$  has two down-edges  $u_iv$  and  $u_iw$  where  $w \notin CCW_b$ . The vertex  $v$  is correctly classified in  $CCU_b$ . If  $w$  is in  $CW_b$  then  $w$  has one down-edge  $wv'$  where  $v' \in U_b$ . Then vertex  $v'$  is closer to  $a$  than  $v$  is, so  $v'$  is closer to  $b$  than  $v$  is, and hence  $v' \in CCU_b$ . Therefore  $v'$  is scanned before  $v$  in the algorithm and hence  $w$  is classified as being in  $CCW_b$ . Since this is not the case we conclude that  $w \notin CW_b$ . Therefore  $u_i$  has two down-edges  $u_iv$  and  $u_iw$  and it holds that  $v \in U_b$  and  $w \notin CW_b$ . By Observation 2,  $u_i \in U_b$ , a contradiction.

We conclude that  $CCU_b = U_b$ . Obviously it follows that  $CCU_a = U_a$  and  $CCF = F$ , completing the proof.  $\square$

With procedure FINDSETS our algorithm reads as follows.

Algorithm MEDIAN ( $a$ )

If  $G$  is not a one vertex graph then

1. choose an edge  $ab$  of  $G$  such that  $b$  has maximal degree among all vertices adjacent to  $a$ ;
2. obtain the sets  $CCU_a, CCU_b$  and  $CCF$  by calling procedure FINDSETS;
3. verify that
  - 3.1  $CCF$  is a matching that defines an isomorphism between  $\langle CCU_a \rangle$  and  $\langle CCU_b \rangle$ ;
  - 3.2 for any  $u \in CCU_a$  (resp.  $CCU_b$ ) and any down-edge  $uw$ ,  $uw \notin CCF$ ,  $w$  is in  $CCU_a$  (resp.  $CCU_b$ );

- 3.3 each vertex  $u \in CCU_a \cup CCU_b \setminus \{a\}$  has at least one down-edge;
  - 3.4 every vertex in  $X := \{w \mid uw \in E(G), u \in CCU_a \text{ (resp. } CCU_b), w \notin CCU_a \cup CCU_b\}$  has exactly one neighbour in  $CCU_a$  (resp.  $CCU_b$ ) and no neighbour in  $CCU_b$  (resp.  $CCU_a$ ).
- if any one of the foregoing conditions is not fulfilled, then REJECT;
- 4. obtain a graph  $G'$  from  $G$  by removing the edges of  $G$  that are in  $CCF$ ;
  - 5. MEDIAN( $b$ ); MEDIAN( $a$ ).

We call the algorithm correct if it accepts median graphs and rejects all non-median graphs. We shall see from the proof of Theorem 3.5 that Step 1 of MEDIAN is not essential for the correctness of the algorithm. However, Step 1 is crucial for the reduction of the complexity from  $O(mn)$  to  $O(m\sqrt{n})$ .

We start with an arbitrary vertex  $v_0$ , and let  $v_1$  be a vertex adjacent to  $v_0$  with the largest degree. We set  $v_0v_1$  to be the first separation edge. Then in the recursive steps, edges  $v_1w_1$  and  $v_0w_0$  will be separation edges, where  $w_1$  and  $w_0$  are neighbours of  $v_1$  and  $v_0$ , respectively, with the largest degree. The order of the choice of the separation edges is implicitly given by Steps 1 and 5 of MEDIAN. As we have already indicated, Step 1 is crucial for the complexity, but Step 5 is also crucial for the correctness.

MEDIAN is started with a vertex  $v_0$ , but first some preprocessing has to be done. It must be tested whether  $G$  is bipartite. We partition the edges into up- and down-edges with respect to  $v_0$ . We point out that this is fixed for all recursive calls. We must also check if  $m \leq cn \log n$ , where  $c$  is a fixed small constant (this bound holds for any subgraph of a hypercube, as proved by Graham [11]).

There are two major questions concerning the correctness of the algorithm. First, if  $ab$  is a parameter for FINDSETS, the procedure is only correct if up- and down-edges refer to the vertex  $a$ . But the edges are partitioned into up- and down-edges with respect to a fixed vertex  $v_0$ . Secondly, if  $CCU_a \neq U_a$  or  $CCU_b \neq U_b$  then  $G$  is not a median graph, but is this recognized by the algorithm?

The following lemma answers the first question.

**Lemma 3.2** *Let  $G$  be a median graph and let  $G'$  be the input graph when  $ab$  is the separation edge. Then  $uv \in E(G')$  is an up-edge or down-edge with respect to  $a$  if and only if it is an up-edge or down-edge, respectively, with respect to  $v_0$ .*

**Proof.** The statement is clearly true if  $a = v_0$ . When MEDIAN is called again with parameter  $a$ , it is still correct. When MEDIAN is called with parameter  $b$  then  $\langle W_b \rangle$  is the input graph because  $G$  is a median graph. We remark that  $G \setminus CCF$  may be connected if  $G$  is not a median graph. Let  $u \in W_b$ . Each shortest path in  $\langle W_b \rangle$  from  $u$  to  $b$  extends to a shortest path from  $u$  to  $a$  in  $G$ . It follows that an up- or down-edge with respect to  $a$  is an up- or down-edge with respect to  $b$ . To complete the proof we use this argument inductively for the recursive calls.  $\square$

**Lemma 3.3** *If for some separation edge  $ab$  removing the edges  $CCF$  does not disconnect the graph, then the algorithm rejects and  $G$  is not a median graph.*

**Proof.** By Lemma 3.1  $G$  is not a median graph.

If the graph is not disconnected by removing the edges of  $CCF$  then there is an edge  $uw \in E(G)$  such that  $u \in U_b$ ,  $w \in U_a$  and  $u \notin CCU_b$  or  $w \notin CCU_a$ . Observe that at the time when  $ab$  is the separation edge the vertex  $a$  has no down-edge.

By the recursive call of the algorithm MEDIAN all successors (with respect to up-edges) will be scanned first. Note that the notion up-edge always refers to the same start vertex. Thus in the course of the algorithm the vertex  $u$  will be classified as being in  $CCU_{a'}$  or  $CCU_{b'}$  for some separation edge  $a'b'$ . In Step 3.2 of the algorithm all down-edges of  $u$  are scanned and  $w$  must be found either in  $CCU_{a'}$  or in  $CCU_{b'}$ . Then all down-edges of  $w$  are scanned until finally the vertex  $a$  is either in  $CCU_{a'}$  or in  $CCU_{b'}$ . If not earlier, at this stage the algorithm rejects because  $a$  has no down-edge and  $a \neq a'$ .  $\square$

We wish to remark that this case can also occur if  $U_a$  and  $U_b$  are both connected. To see this, consider a 3-cube and remove one vertex. Add at least one pendant edge to a vertex of degree two and start the algorithm with it. The  $CCF$  does not disconnect.

**Lemma 3.4** *If for some separation edge  $ab$ ,  $CCU_b \neq CU_b$  or  $CCU_a \neq CU_a$  then the algorithm rejects and  $G$  is not a median graph.*

**Proof.** It follows by Lemma 3.1 that  $G$  is not a median graph. We may without loss of generality assume that  $ab$  is the first separation edge such that  $CCU_b \neq CU_b$  or  $CCU_a \neq CU_a$ . Assume furthermore that  $u$  is the first vertex that is classified such that  $CCU_b \neq CU_b$  or  $CCU_a \neq CU_a$ .

*Case 1:  $CCU_b \neq CU_b$ .*

We distinguish two subcases.

*Case 1.1:  $u \in CCU_b$  and  $u \notin CU_b$ .*

In this subcase  $u \in CW_b$ . But then there is no down-edge  $uw$  such that  $w \in CCU_a$  and the graph does not pass the isomorphism test.

*Case 1.2:  $u \in CU_b$  and  $u \notin CCU_b$ .*

In this case  $u \in CCW_b$ . Clearly  $uw \notin CCF$ . Thus, there is a down-edge  $uw \in E(G)$  such that  $w \in U_a$ . Removing  $CCF$  does not disconnect the graph, because  $b \dots uw \dots a$  is a path in  $G \setminus CCF$ . By Lemma 3.3 the algorithm rejects.

*Case 2:  $CCU_b = CU_b$  and  $CCU_a \neq CU_a$ .*

In this case the isomorphism test fails and the algorithm rejects.  $\square$

**Theorem 3.5** *MEDIAN correctly recognizes median graphs. In other words, it accepts median graphs and rejects all non-median graphs.*

**Proof.** For all graphs accepted we have  $CCU_b = CU_b$  and  $CCU_a = CU_a$  by Lemma 3.4. Thus, for accepted graphs we have  $CCU_b \subseteq U_b$  and  $CCU_a \subseteq U_a$ . If  $CCU_b \neq U_b$ , then  $CCF$

would not disconnect, but such graphs are rejected by Lemma 3.3. Thus  $CCU_b = U_b$ , and since  $CCF$  disconnects it has to be equal to  $F$  and  $CCU_a$  must be equal to  $U_a$ .

Therefore, the only graphs accepted satisfy  $CCU_b = U_b$ ,  $CCU_a = U_a$  and  $CCF = F$  for all separation edges  $ab$ .

In other words, for all accepted graphs condition (i) of Theorem 2.2 holds. Moreover, conditions (ii) and (iii) of the same theorem are verified in Steps 3.2 and 3.4, respectively, of MEDIAN.

Hence, a graph accepted by the algorithm satisfies the conditions of Theorem 2.2, i.e. it is a median graph.

Conversely, let  $G$  be a median graph. Then by Lemma 3.1  $CCU_b = U_b$ ,  $CCU_a = U_a$  and  $CCF = F$ . Thus, conditions 3.1, 3.2 and 3.4 of MEDIAN hold for median graphs by conditions (i), (ii) and (iii) of Theorem 2.2. Condition 3.3 of MEDIAN holds for median graphs by Lemma 3.4.  $\square$

If one deletes Step 3.4 of the algorithm a class of graphs is accepted that contains all median graphs. We can actually extend the algorithm to obtain an embedding of a given median graph in a hypercube. To this end we observe that the edges of an  $n$ -cube can be canonically colored by  $n$  colors by coloring any edge whose endvertices differ in the  $i$ -th bit with color  $i$ . (E.g. the edge  $[001101, 001010]$  of the 6-cube would be given color 4.) Then it is easy to see that all edges in sets  $F_{ab}$  have to have the same color and that edges of  $U_a$  and  $U_b$  which are matched by  $F_{ab}$  have to have the same color too. Thus, first we color all  $F_{ab}$ . Since every edge is in exactly one  $F_{ab}$ , this colors all edges uniquely. Now we consider all pairs  $e, f$  of edges matched by some  $F_{ab}$  (there are at most  $m \log n$  such pairs) and if  $e$  and  $f$  do not have the same color yet, we merge colors. Thus, we obtain a coloring of the given median graph which defines an isometric embedding into a hypercube.

It is not so easy to show that the same procedure, applied to all nonmedian graphs that are accepted by the algorithm after deletion of Step 3.4, also isometrically embeds them into a hypercube.

## 4 Analysis of the algorithm

In this section we are going to show that MEDIAN runs in  $O(n^{\frac{3}{2}} \log n)$  time.

**Lemma 4.1** *Let  $G$  be a median graph and let  $v \in V(G)$  have  $k$  down-edges with respect to a vertex  $v_0$ . Then  $|G| \geq 2^k$ .*

**Proof.** By Theorem 2.1 (ii),  $G$  can be isometrically embedded in a hypercube  $Q_s$  for some  $s \geq 1$ . Let  $\ell$  be a corresponding labeling, i.e., for any  $u, v \in V(G)$ ,  $d_G(u, v) = H(\ell(u), \ell(v))$ , where  $H(x, y)$  denotes the Hamming distance of the binary strings  $x$  and  $y$ . We may assume that  $\ell(v_0) = 00 \dots 0$ .

As  $v$  has  $k$  down-edges with respect to  $v_0$ ,  $\ell(v)$  differs from  $\ell(v_0)$  in at least  $k$  bits. (Note that  $k \leq s$ .) Therefore  $\ell(v)$  contains at least  $k$  non-zero components and we may without loss of generality assume that the first  $k$  bits of  $\ell(v)$  equal 1. Let  $vu_i$ ,  $1 \leq i \leq k$ , be down-edges of  $v$  (with respect to  $v_0$ ), where the  $i$ -th bit of  $\ell(u_i)$  is 0.

To complete the proof we claim that for each binary string  $x$  that agrees with  $\ell(v)$  on the components  $k + 1, k + 2, \dots, s$ , there is a vertex  $u \in V(G)$  with  $\ell(u) = x$ . Let  $t$  be the number of zero's in the first  $k$  components of  $x$ . The claim will be proved by induction on  $t$ . For  $t = 0$  and  $t = 1$  the claim clearly holds. Let  $x$  be a string with zeros in the components  $i_1, i_2, \dots, i_t, 2 \leq t \leq k, i_t \leq k$ . By the induction hypothesis, there exists a vertex  $w \in V(G)$  such that  $\ell(w)$  has zeros in the components  $i_1, i_2, \dots, i_{t-1}$ . Consider the vertices  $v_0, w$  and  $u_{i_t}$ . Their unique median has the label in which a component is obtained as a majority over the corresponding components in  $\ell(v_0), \ell(w)$  and  $\ell(u_{i_t})$  (cf. [19, 8]). We conclude that  $x$  is the median of  $\ell(v_0), \ell(w)$  and  $\ell(u_{i_t})$ , which proves the claim.  $\square$

Note that the proof of Lemma 4.1 implies that the hypercube  $Q_k$  is a subgraph of the interval  $I(v, v_0)$ .

**Corollary 4.2** *Let  $G$  be a median graph. Then no vertex has more than  $\log n$  down-edges with respect to any vertex.*

As mentioned in Section 3, the calculation of up- and down-edges with respect to the vertex  $v_0$  must be done before MEDIAN is called. For the desired complexity of our algorithm, additional preprocessing has to be done: a check that the condition of Corollary 4.2 holds. We are now ready to analyse the time complexity of the algorithm MEDIAN.

We first analyse the time complexity of the procedure FINDSETS. Observe that only edges adjacent to vertices in  $CCU_b$  are scanned. By Step 2.2 exactly one down-edge of every up-neighbor  $u$  of  $v \in CCU_b$  is considered. Therefore the time complexity of one call of FINDSETS is in  $O(\sum_{v \in CCU_b} d(v))$ . Observe furthermore that whenever a vertex  $v$  is in  $CCU_b$  then a down-edge of this vertex is removed. Since each vertex has at most  $\log n$  down-edges, a vertex  $v$  can be in  $CCU_b$  at most  $\log n$  times for a fixed vertex  $b$ . Therefore the overall complexity of the procedure FINDSETS is  $O(m \log n)$ .

Now consider the complexity of the algorithm MEDIAN. Assuming a careful implementation, Steps 3.1 - 3.3 can be done in time linear in the number of edges. With the same argument as above the overall complexity of these steps is  $O(m \log n)$ . To test Step 3.4 for  $CCU_b$ , one has to go through the adjacency list of vertices in  $CCU_b$  and mark vertices that are not in  $CCU_b$  or  $CCU_a$ . If such a vertex is marked more than once the test fails and REJECT is called. Again, with the argument from above this takes  $O(m \log n)$  time.

The most time-consuming part is Step 3.4 for  $CCU_a$ . Each call can be done in time  $O(\sum_{v \in CCU_a} d(v))$ . However, we cannot use Lemma 4.1 because whenever a vertex is in  $CCU_a$  an up-edge is removed, but not a down-edge. A vertex might have  $O(n)$  up-edges. We wish to show that a vertex  $v$  can be in  $CCU_a$  at most  $\sqrt{n}$  times.

Let  $G$  be a graph and consider the separation edges chosen by the algorithm MEDIAN. These edges define a spanning tree which we call *Tree of Separation Edges* (TSE). If  $G$  is a median graph, then TSE is a spanning tree of  $G$ . We call a vertex  $v'$  the (unique) *predecessor* of a vertex  $v$  if  $v'v$  is a separation edge. For a vertex  $v$  let  $f(v) := k$  if there are exactly  $k$  separation edges  $ab, a \neq v$ , such that  $v \in CCU_a$ .

**Lemma 4.3** *Let  $v \in V(G)$  and let  $f(v) = k$ . Then  $|V(G)| \geq \frac{1}{4} \cdot \binom{k}{2}$ .*

**Proof.** Observe first that whenever  $v \in CCU_a$ , an edge adjacent to  $v$  is removed. Therefore  $d(v) \geq k$ . Let  $v'$  be the predecessor of  $v$  and let  $f(v') = k'$ .

*Claim:*  $d(v') \geq k$  and  $v'$  is adjacent to at least  $k - k'$  vertices of degree at least  $k$ .

The claim clearly holds for  $k = k'$ . Assume next  $k' < k$  and consider the situation when for the first time a separation edge  $ab$  is chosen such that  $a = v'$ . Observe that from that point on,  $v'$  no longer occurs in  $CCU_{a'}$  for some separation edge  $a'b'$ . Furthermore, at that time  $v$  has occurred at most  $k'$  times in  $CCU_{a'}$  for some separation edge  $a'b'$ , because Step 3.2 of MEDIAN implies that whenever  $v \in CCU_{a'}$  then  $v' \in CCU_{a'}$ . Whenever  $v$  occurs from now on in  $CCU_a$  for some separation edge  $ab$ , we have  $a' = v'$  and  $b \neq v$ . This implies, by the design of our algorithm, that  $d(b) \geq d(v)$ . Therefore  $v'$  must have at least  $k - k'$  adjacent vertices of degree at least  $d(v)$ . Thus the claim holds.

So far we have encountered at least  $(k - k') \cdot k \geq \sum_{i=k'}^k i$  edges. Applying our Claim inductively along the unique path from  $v$  to  $v_0$  in TSE and using the fact that  $f(v_0) = 0$  we obtain  $|E(G)| \geq \sum_{i=1}^k i = \binom{k}{2}$ .

To obtain a similar bound for the number of vertices, consider the vertex  $v'$  and its  $k - k'$  adjacent vertices  $u_1, u_2, \dots, u_{k-k'}$ . By the claim,  $d(u_i) \geq k$ ,  $i = 1, 2, \dots, k - k'$ . Furthermore, a vertex (different from  $v'$ ) has at most two adjacent vertices in  $\{u_1, u_2, \dots, u_{k-k'}\}$ , because otherwise we encounter an induced  $K_{2,3}$ . Thus there are at least  $\frac{1}{2}k(k - k')$  different vertices adjacent to the vertices  $u_i$ . By summing up the number of vertices along the path from  $v$  to  $v_0$  and observing that each vertex is counted at most twice we get  $|V(G)| \geq \frac{1}{4} \cdot \binom{k}{2}$ .  $\square$

**Corollary 4.4** *For any vertex  $v \in V(G)$ ,  $f(v) \in O(\sqrt{n})$ .*

For a fixed vertex  $v$  each time  $d(v)$  edges have to be considered in Step 3.4 for  $CCU_a$ . Therefore Corollary 4.4 implies that the overall complexity is  $O(m\sqrt{n}) = O(n^{\frac{3}{2}} \log n)$ , which is then also the complexity of MEDIAN:

**Theorem 4.5** *MEDIAN runs in  $O(n^{\frac{3}{2}} \log n)$  time.*

We have seen that only Step 3.4 of the algorithm is of complexity  $O(m\sqrt{n})$  while all the other steps are of complexity  $O(m \log n)$ . Furthermore, if we run the algorithm without Step 3.4 then 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.) These observations give us:

**Theorem 4.6** *A median graph can be isometrically embedded into a hypercube in  $O(m \log n)$  time.*

We conclude the paper by showing that the choice in which separation edges are selected is essential for the running time of our algorithm. In fact, an arbitrary sequence of separation edges would lead to an  $O(n^2 \log n)$  algorithm - the complexity of the algorithms due to Jha and Slutzki [15, 16].

Let  $G_k$  be a graph with the vertex set

$$V(G_k) = \{u, v, u_1, u_2, \dots, u_k, v_1, v_2, \dots, v_k\},$$

and edges

$$E(G_k) = \{uv\} \cup \{uu_i, vv_i, u_i v_i \mid i = 1, 2, \dots, k\}.$$

The graph  $G_5$  is shown on Fig. 1.

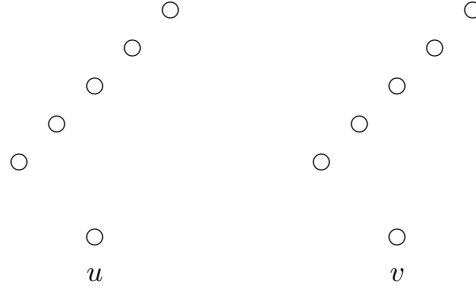


Figure 1: The graph  $G_5$

If  $u = v_0$  and the edge  $uv$  is scanned at the end, then  $f(v) = k$ . The first time one has to go through  $k - 1$  edges, then  $k - 2$ , and so on. As  $|V(G_k)| = n = 2(k + 1)$ ,  $\binom{n}{2}$  edges are scanned in the convexity test (Step 3.4 of MEDIAN). However, if  $uv$  were chosen as the first separation edge, only  $O(n)$  edges are scanned.

## Acknowledgement

We wish to thank the referee for his careful reading of the manuscript and his invaluable remarks that helped to improve the paper considerably.

## References

- [1] S.P. Avann, Metric ternary distributive semi-lattices, Proc. Amer. Math. Soc., 12 (1961), 407–414.
- [2] F. Aurenhammer, M. Formann, R. Idury, A. Schäffer and F. Wagner, Faster isometric embeddings in products of complete graphs, Disc. Appl. Math., 52 (1994), 17–28.
- [3] F. Aurenhammer and J. Hagauer, Recognizing binary Hamming graphs in  $O(n^2 \log n)$  time, Lecture Notes in Computer Science, Vol. 484 (Springer, New York, 1991) 90–98. Also Math. Systems Theory, 28 (1995), 387–396.
- [4] H.J. Bandelt, Retracts of hypercubes, J. Graph Theory, 8 (1984), 501–510.
- [5] H.J. Bandelt and J.P. Barthélemy, Medians in median graphs, Discrete Appl. Math., 8 (1984), 131–142.

- [6] H.J. Bandelt and H.M. Mulder, Infinite median graphs, (0,2)-graphs, and hypercubes, *J. Graph Theory*, 7 (1983), 487–497.
- [7] 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.
- [8] F.R.K. Chung, R.L. Graham, and M.E. Saks, A dynamic location problem for graphs, *Combinatorica*, 9 (1989), 111–132.
- [9] D. Duffus and I. Rival, Graphs orientable as distributive lattices, *Proc. Amer. Math. Soc.*, 88 (1983), 197–200.
- [10] T. Feder, Product graph representations, *J. Graph Theory* 16, (1992), 467–488.
- [11] R.L. Graham, On primitive graphs and optimal vertex assignments, *Ann. New York Acad. Sci.*, 175 (1970), 170–186.
- [12] W. Imrich and S. Klavžar, A simple  $O(mn)$  algorithm for recognizing Hamming graphs, *Bull. Inst. Comb. Appl.*, 9 (1993), 45–56.
- [13] W. Imrich and S. Klavžar, On the complexity of recognizing Hamming graphs and related classes of graphs, *European J. Combin.*, 17 (1996), 209–221.
- [14] W. Imrich and S. Klavžar, A convexity lemma and expansion procedures for bipartite graphs, manuscript, 1997.
- [15] P.K. Jha and G. Slutzki, On  $O(n^2 \log n)$  algorithm for recognizing median graphs, manuscript, 1989.
- [16] P.K. Jha and G. Slutzki, Convex-expansions algorithms for recognizing and isometric embedding of median graphs, *Ars Combin.*, 34 (1992), 75–92.
- [17] F.R. McMorris, H.M. Mulder and F.S. Roberts, The median procedure on median graphs, Report 9413/B, Econometric Institute, Erasmus University Rotterdam.
- [18] H.M. Mulder, The structure of median graphs, *Discrete Math.*, 24 (1978), 197–204.
- [19] H.M. Mulder, The Interval Function of a Graph, *Mathematical Centre Tracts 132*, Mathematisch Centrum, Amsterdam, 1980.
- [20] H.M. Mulder,  $n$ -Cubes and median graphs, *J. Graph Theory*, 4 (1980), 107–110.
- [21] H.M. Mulder and A. Schrijver, Median graphs and Helly hypergraphs, *Discrete Math.*, 25 (1979), 41–50.
- [22] L. Nebeský, Median graphs, *Comment. Math. Univ. Carolinae*, 12 (1971), 317–325.