Computing median and antimedian sets in median graphs*

Kannan Balakrishnan
Department of Computer Applications
Cochin University of Science and Technology, Cochin-22, India
bkannan@cusat.ac.in

Boštjan Brešar
Faculty of Electrical Engineering and Computer Science
University of Maribor
Smetanova 17, 2000 Maribor, Slovenia
bostjan.bresar@uni-mb.si

Manoj Changat
Department of Futures Studies
University of Kerala, Trivandrum-695034, India
mchangat@gmail.com

Sandi Klavžar
Department of Mathematics and Computer Science
FNM, University of Maribor
Gosposvetska 84, 2000 Maribor, Slovenia
sandi.klavzar@uni-mb.si

Matjaž Kovše
Institute of Mathematics, Physics and Mechanics,
Jadranska 19, 1000 Ljubljana, Slovenia
matjaz.kovse@gmail.com

Ajitha R. Subhamathi
Department of Futures Studies
University of Kerala, Trivandrum-695034, India
ajithars@gmail.com

May 27, 2008

[*] Work supported by the Ministry of Science of Slovenia and by the Ministry of Science and Technology of India under the bilateral India-Slovenia grants BI-IN/06-07-002 and DST/INT/SLOV-P-03/05, respectively.

Abstract

The median (antimedian) set of a profile $\pi = (u_1, \ldots, u_k)$ of vertices of a graph G is the set of vertices x that minimize (maximize) the remoteness $\sum_i d(x, u_i)$. Two algorithms for median graphs G of complexity $O(n\,\mathrm{idim}(G))$ are designed, where n is the order and $\mathrm{idim}(G)$ the isometric dimension of G. The first algorithm computes median sets of profiles and will be in practice often faster than the other algorithm which in addition computes antimedian sets and remoteness functions and works in all partial cubes.

Keywords:

1 Introduction

The problems of locating medians and antimedians of profiles are natural generalizations of facility location problem in graphs. Given a profile (that is, a multiset of vertices) on a graph/network one wants to locate vertices whose remoteness, that is, the sum of distances to the vertices of the profile, is minimum or maximum. This is a model for those practical problems where vertices of the profile present customers and the resulting median (resp. antimedian) set consists of optimal locations of desirable (resp. undesirable) facilities in the network.

The median problem for profiles on graphs was considered by many authors; see, for example, [1, 3, 4, 6, 18]. On the other hand, not much work has been done so far for the general antimedian problem for profiles on graphs. To be exact, there are some papers that deal with the special case of this problem—the obnoxious facility location problem on graphs—where profiles always coincide with the vertex set of a graph; see [7, 9, 20, 22, 24]. We think that the problem of antimedian for profiles could be of similar interest for applications, as well as theoretically, being the opposite extremum to the median problem.

Median graphs form a closely investigated and well understood class of graphs; see [16] for a survey, in particular for their role in location theory. Many important and interesting classes of graphs are median, let us just mention trees, hypercubes, square grids, graphs of acyclic cubical complexes, simplex graphs, and Fibonacci and Lucas cubes. Most of these classes have been considered from the algorithmic point of view. Median graphs themselves can be recognized in subquadratic time [12, 14], in fact, their recognition complexity is essentially the same as the complexity of recognizing triangle-free graphs [15], see also [10]. On the other hand, the recognition complexity can be reduced to $O(m \log n)$ for acyclic cubical complexes [13] and for Fibonacci cubes [21]. Here and later n and m denote the number of vertices and edges of a given graph. For additional fast algorithms on classes of median graphs see [8].

The main purpose of this paper is to present efficient algorithm(s) for computing median and antimedian sets of profiles on median graphs. As the main tool we use isometric (that is, distance preserving) embeddings of median graphs into hypercubes. In the next section we fix notation, state necessary definitions, and present a simple algorithm for computing median and antimedian sets in general graphs. In the subsequent section we prove that, given a graph G, embedded via a weak contraction into a graph H, the median set of a profile on G can be obtained from the corresponding median set in H. We elaborate this idea in Section 4 for the case when a median graph G is isometrically embedded into a hypercube to obtain a fast algorithm for computing median sets in median graphs. (Unfortunatelly, a similar theorem does not hold for antimedians.) We follow with another algorithm in Section 5 that in addition computes antimedian sets and the remoteness functions within the same time and works in arbitrary isometric subgraphs of hypercubes. In practice, however, the first algorithm will often be faster. In the final section we consider possible generalizations and variations of our approach.

2 Preliminaries

In this paper we consider simple, undirected, connected graphs. The *distance* considered is the usual shortest path distance d. For a connected graph G and subsets of vertices $X, Y \subseteq V(G)$ we will write $d(X, Y) = \min\{d(x, y) \mid x \in X, y \in Y\}$. In particular, for a vertex u of G and a set of vertices X we have $d(u, X) = \min\{d(u, x) \mid x \in X\}$. The distance $d(x, \pi)$ between the vertex x and the profile π is defined analogously.

A *profile* $\pi = (x_1, \ldots, x_k)$ on a graph G is a finite sequence of vertices of G. Note that in a profile a vertex may be repeated. Given a profile π on G and a vertex u of G the *remoteness* $D(u, \pi)$ (see [17]) is

$$D(u, \pi) = \sum_{x \in \pi} d(u, x)\,.$$

The vertex u is called a *median (antimedian) vertex for* π if $D(u, \pi)$ is minimum (maximum). The *median (antimedian) set* $M(\pi, G)$ $(AM(\pi, G))$ of π in G is the set of all median (antimedian) vertices for π.

A vertex x is a *median* of a triple of vertices u, v and w if $d(u, x) + d(x, v) = d(u, v)$, $d(v, x) + d(x, w) = d(v, w)$ and $d(u, x) + d(x, w) = d(u, w)$. A (connected) graph G is a *median graph* if every triple of its vertices has a unique median. The *hypercube* or *d-cube* Q_d, $d \geq 1$, is the graph with vertex set $\{0, 1\}^d$, two vertices being adjacent if the corresponding tuples differ in precisely one position. A vertex u of Q_d will be written in its coordinate's form as $u = u^{(1)} \ldots u^{(d)}$. Note that the distance in Q_n between two vertices is the number of coordinates in which they differ. The latter distance is known as the *Hamming distance*.

A subgraph H of a (connected) graph G is an *isometric subgraph* if $d_H(u, v) = d_G(u, v)$ holds for any vertices $u, v \in H$. Let G be an isometric subgraph of some hypercube (such graphs are also called *partial cubes*). The smallest integer d such that G is an isometric subgraph of Q_d is called the *isometric dimension* of G and denoted $\mathrm{idim}(G)$. An important structural result due to Mulder [19] asserts that every median graph G can be isometrically embedded in a hypercube such that the median set of every profile π of cardinality three in G on the hypercube coincides with the median set of π in G.

The median and the antimedian set of a profile π on a connected graph G can be obtained by the following straightforward algorithm. For each vertex $x \in \pi$ (of course, taking into account multiple occurrences of x as well), perform a BFS from x. In this way the distances from x to all vertices of G are obtained and summed up along the way. After this is done, the sum of the distances to the vertices of the profile is obtained, and then the vertices with minimum and maximum sum are easily determined. The time complexity of this algorithm is the number of edges of G (the time complexity for performing a BFS) times the number of different vertices that appear in π. Although this is quite efficient, it may be the case that faster algorithms are required, in particular in large networks when the results are needed in real-time.

3 Median sets with respect to isometric embeddings

Recall that median graphs are isometric subgraphs of hypercubes [19]. The main idea of this paper is to compute median and antimedian sets in median graphs using these embeddings. In this section we give a fundamental theoretical observation for this purpose which holds in a more general setting.

The concept of a weak contraction was used by Feder [11, Theorem 6.28] to prove a fixed box theorem for the distance center of subgraphs of Cartesian product graphs. (The distance center of G is the median of the profile consisting of the vertex set of G.) Let G be a subgraph of H. A mapping $f\colon V(H) \to V(G)$ is called a *weak contraction* (or *weakly nonexpansive map)* if for all $u \in V(H)$ and all $v \in V(G)$ we have $d_G(f(u), v) \leq d_H(u, v)$. Note that for $v \in V(G)$ we have $f(v) = v$ because $d_G(f(v), v) \leq d_H(v, v) = 0$. Therefore, for any $u, v \in V(G)$,

$$d_G(u, v) = d_G(f(u), v) \leq d_H(u, v)\,.$$

Hence, since G is a subgraph of H, $d_G(u, v) = d_H(u, v)$, that is, every weak contraction is an isometry. So the condition $d_G(f(u), v) \leq d_H(u, v)$ is equivalent to $d_H(f(u), v) \leq d_H(u, v)$.

Theorem 3.1 *Let G be a subgraph of H such that there exists a weak contraction f from H to G. Then for any profile π on G, $M(\pi, G) = M(\pi, H) \cap V(G)$.*

Proof. Note that we only need to prove that $M(\pi, H) \cap V(G) \neq \emptyset$. Suppose $u \in V(H) \setminus V(G)$ such that $u \in M(\pi, H)$. We claim that $f(u) \in M(\pi, H)$ (since

$f(u) \in V(G)$ this will be sufficient). Clearly,

$$D_H(u,\pi) = \sum_{x\in\pi} d_H(u,x) \geq \sum_{x\in\pi} d_G(f(u),x) = \sum_{x\in\pi} d_H(f(u),x) = D_H(f(u),\pi)$$

which readily implies $f(u) \in M(\pi, H)$. □

It is easily seen that every (weak) retract is a weak contraction, hence the above theorem holds also for these two special cases. In particular, it is well-known that median graphs are precisely retracts of hypercubes [2], hence we deduce

Corollary 3.2 *Let G be a median graph, isometrically embedded into a hypercube H. Then for any profile π on G, $M(\pi,G) = M(\pi,H) \cap V(G)$.*

Not all partial cubes can be obtained as weak contractions of hypercubes. For instance, it is easy to see that the cycle C_6 is not a weak contraction of the hypercube Q_3. We wonder whether only median graphs have this property.

In the formula of Theorem 3.1 it is necessary to make the intersection with $V(G)$. For instance, let $\pi = (u,v)$, then the median set of π consists of all vertices lying on shortest u,v-paths which can in Q_d yield a larger set than in G. As the smallest example consider the path on three vertices embedded in Q_2. For another small example see Fig. 1, where G is the graph induced with black vertices. Then $M(\pi,G) = V(G)$ but $M(\pi,Q_3) = V(Q_3)$.

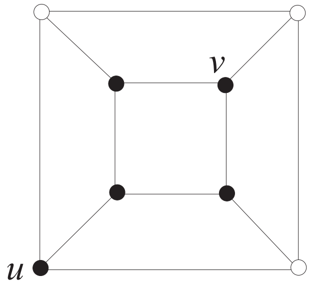

Figure 1: A small example

Note that the analogue of Theorem 3.1 for antimedian sets does not hold in general. In fact, in many cases the corresponding intersection will be empty which makes this problem more difficult as the median problem.

4 Fast computation of median sets in median graphs

In this section we design an efficient algorithm for computing the median set of a profile on a median graph by applying Corollary 3.2. Thus we first have a closer look to median (and antimedian) sets in hypercubes. Part of this might be a folklore.

Let $\pi = (x_1, \dots, x_k)$ be a profile on Q_d. For $i = 1, \dots, k$ let $n_0^{(i)}$ and $n_1^{(i)}$ be the number of vertices from π with the ith coordinate equal 0 and 1, respectively. More formally,

$$n_0^{(i)}(\pi) = |\{x \in \pi \mid x^{(i)} = 0\}|$$

and

$$n_1^{(i)}(\pi) = |\{x \in \pi \mid x^{(i)} = 1\}| \,.$$

Define Majority(π) as the set of vertices $u = u^{(1)} \dots u^{(d)}$ of Q_d, where

$$u^{(i)} \begin{cases} = 0; & n_0^{(i)}(\pi) > n_1^{(i)}(\pi), \\ = 1; & n_0^{(i)}(\pi) < n_1^{(i)}(\pi), \\ \in \{0,1\}; & n_0^{(i)}(\pi) = n_1^{(i)}(\pi) \,. \end{cases}$$

We say that vertices $u \in$ Majority(π) are obtained by the *majority rule*. Minority(π) and the *minority rule* are defined analogously.

Note that it follows immediately from the definitions that if $|\pi|$ is odd then $|\text{Majority}(\pi)| = 1$ and $|\text{Minority}(\pi)| = 1$.

Proposition 4.1 *Let $\pi = (x_1, \dots, x_k)$ be a profile on Q_d. Then $M(\pi, Q_d) =$* Majority(π) *and $AM(\pi, Q_d) =$* Minority(π).

Proof. Let $u \in$ Majority(π) and let w be an arbitrary vertex of Q_d. For $b, b' \in \{0,1\}$ set $\delta(b,b') = 0$ if $b = b'$ and $\delta(b,b') = 1$ otherwise. Then,

$$\begin{aligned} D(u,\pi) &= \sum_{i=1}^{k} d(u, x_i) = \sum_{i=1}^{k} \sum_{j=1}^{d} \delta(u^{(j)}, x_i^{(j)}) \\ &= \sum_{j=1}^{d} \sum_{i=1}^{k} \delta(u^{(j)}, x_i^{(j)}) \\ &\leq \sum_{j=1}^{d} \sum_{i=1}^{k} \delta(w^{(j)}, x_i^{(j)}) \\ &= D(w,\pi) \,. \end{aligned}$$

Note that the above inequality holds by the construction of the Majority(π). Moreover, equality holds if and only if $w \in$ Majority(π). We conclude that $M(\pi, Q_d) =$ Majority(π).

The arguments for the antimedian set are analogous. □

Combining Theorem 3.1 with Proposition 4.1 we get the following algorithm for computing median sets in median graphs.

Algorithm 1

```
Input:   A median graph $G$ isometrically embedded into a hypercube $Q$. A profile $\pi$.
Output:  $M(\pi, G)$.
Step 1:  Find $M(\pi, Q)$ using the majority rule.
Step 2:  Compute $M(\pi, G) = M(\pi, Q_d) \cap V(G)$.
```

Theorem 4.2 *Algorithm 1 correctly computes the median set $M(\pi, G)$ in a median graph G and can be implemented in $O(n\,\mathrm{idim}(G))$ time.*

Proof. Correctness of the algorithm follows from Theorem 3.1 and Proposition 4.1.

For the time complexity we first note that if π contains multiple occurrences of the same vertex it is not difficult to modify the algorithm such that this vertex is considered only once. (We do not list multiple occurrences of it but only count its frequency when computing $M(\pi, Q)$ using the majority rule.)

Since every vertex of π has $\mathrm{idim}(G)$ coordinates, Step 1 can be performed in $O(n\,\mathrm{idim}(G))$ time. Along with Step 1 we can perform Step 2 as follows. As soon as we determine the majority in the ith coordinate, we mark all the vertices of G that have the ith coordinate different from the majority as non-median. (Note that if the frequency of 0's and 1's in the ith coordinate is equal, we do nothing.) At the end we are left with the median set. Altogether we need $O(n\,\mathrm{idim}(G))$ operations. □

Algorithm 1, together with the general BFS approach for computing median sets (as described in Section 2) yields an algorithm of complexity

$$\min\{O(n\,\mathrm{idim}(G)), O(m\,|\pi \cap V(G)|)\}$$

for computing median sets in median graphs. Since in these graphs $m = O(n \log n)$ (see [14]), Algorithm 1 will be better than the general algorithm if $\mathrm{idim}(G)$ is smaller than $|\pi \cap V(G)|\ \log n$. In many practical situations this is indeed the case, as for instance when the median graph G is close to a hypercube structure. (Recall that for a hypercube Q_d we have $\mathrm{idim}(Q_d) = d = \log n$, where $n = |Q_d|$.)

In practice Algorithm 1 can perform less than $n\,\mathrm{idim}(G)$ operations if $M(\pi, Q_d)$ is relatively small and we are somehow lucky with the coordinatization of the median graph G. Namely, when implementing Step 2 by marking vertices as non-median it may happen that almost all vertices are marked as non-median after only a few coordinates were checked. By putting each such vertex at the end of the list, we then need to examine only vertices that are not marked (in the ideal case only those that are in $M(\pi, G)$). Denoting the number of different elements in the profile π with $\|\pi\|$, in such events the complexity of Algorithm 1 is $O(\|\pi\|\,\mathrm{idim}(G))$, which is better than $O(n\,\mathrm{idim}(G))$.

5 A more general fast algorithm

In this section we give another algorithm for computing median sets in median graphs of the same complexity $O(n\,\mathrm{idim}(G))$. Its advantage is that it can also be used for computing the antimedian sets and values of $D(x,\pi)$ for all x and any profile on a (median) graph. In addition, it works in arbitrary isometric subgraphs of hypercubes. Its disadvantage is that the number of its operations is fixed (regardless of the coordinatization of the graph, the choice of the profile, etc.), which means that in practice Algorithm 1 will in many cases run faster.

Note that when performing the majority rule values $n_0^{(j)}(\pi)$ and $n_1^{(j)}(\pi)$ are computed, and then compared. We may store these values in two vectors, that is, for any profile π on a median graph G with $k = \mathrm{idim}(G)$, let

$$\overrightarrow{0}(\pi) = (n_0^{(1)}(\pi), \ldots, n_0^{(k)}(\pi)),$$

and

$$\overrightarrow{1}(\pi) = (n_1^{(1)}(\pi), \ldots, n_1^{(k)}(\pi)).$$

For a vertex x of G we define the vector $\overrightarrow{d}(x,\pi)$ as follows:

$$\overrightarrow{d}_j(x,\pi) = \begin{cases} \overrightarrow{0}_j(\pi); & x^{(j)} = 1, \\ \overrightarrow{1}_j(\pi); & x^{(j)} = 0, \end{cases}$$

for $j = 1, \ldots, k$. It is easy to see, using the definition of the Hamming distance, that for any vertex $x \in V(G)$ we have

$$D(x,\pi) = \sum_{j=1}^{k} \overrightarrow{d}_j(x,\pi)\,. \qquad (1)$$

We derive the following algorithm for computing $D(x,\pi)$ for any vertex of a partial cube G.

Algorithm 2

```
Input:  A graph $G$, isometrically embedded into a hypercube $Q$. A profile $\pi$.
Output: $D(x,\pi)$ for all vertices $x \in V(G)$, $M(\pi,G)$ and $AM(\pi,G)$.
Step 1: Using the majority rule in $Q$, determine $\overrightarrow{0}(\pi)$ and $\overrightarrow{1}(\pi)$.
Step 2: For every $x \in V(G)$ compute $D(x,\pi)$ using (1).
Step 3: Determine $M(\pi,G)$ and $AM(\pi,G)$ as the vertices with the smallest (largest)
        $D(x,\pi)$.
```

Theorem 5.1 *Let x be a vertex of a graph G, and let π be a profile on G. Then Algorithm 2 correctly computes $D(x,\pi)$, the median set and the antimedian set of π, and can be implemented in $O(n\,\mathrm{idim}(G))$ time.*

Proof. Observations preceding the algorithm imply the correctness of the algorithm.

For the time complexity observe first that Step 1 is the same as Step 1 of Algorithm 1 except that in the present algorithm vectors $\overrightarrow{0}(\pi)$ and $\overrightarrow{1}(\pi)$ are explicitly determined and saved. Hence Step 1 can be performed in $O(n\,\mathrm{idim}(G))$ time. For Step 2 we need $O(n\,\mathrm{idim}(G))$ operations since for every vertex of G we check each of its coordinates, and this check is done in a constant time (upon the value of $x^{(j)}$ decide to choose either $\overrightarrow{0_j}$ or $\overrightarrow{1_j}$, and add this number to the sum in (1)). Along with Step 2 we can perform Step 3 in such a way that vertices x with current largest and smallest value of $D(x,\pi)$ are kept during the entire algorithm. This yields a constant number of operations while processing each vertex. Hence the overall time complexity is $O(n\,\mathrm{idim}(G))$. □

We conclude this section with a word on the preprocessing required by Algorithms 1 and 2. In practice we suppose that a graph/network is already given, so that we only need to embed it into a hypercube. It was proved in [12] that this can be done in $O(m \log n)$ time for semi-median graphs (a class of partial cubes that include median graphs). On the other hand, the fastest known algorithm for recognizing median graph is presented in [14] and is of complexity $O((m \log n)^{2\omega/(\omega+1)})$, where ω is the exponent of matrix multiplication.

6 Concluding remarks

Median graphs have many natural generalizations; one such well studied class is the family of quasi-median graphs; see [5]. Among many similarities with median graphs, quasi-median graphs admit isometric embeddings into Hamming graphs. Moreover, they are weak retracts of Hamming graphs [23], hence Theorem 3.1 can be applied for quasi-median graphs as well. By using a natural extension of majority rule to Hamming graphs, one can easily generalize Algorithm 1 to work for quasi-median graphs.

Step 1 of this algorithm indeed directly extends from hypercubes to Hamming graphs. Having in mind that the vertices of a Hamming graph are t-tuples (whose coordinates are taken from $\{0,1,\ldots,k_i\}$, respectively), the majority/minority rule is generalized to the set of integers that are most/least frequent in a given coordinate. This implies that in a Hamming graph an (anti)median set induces a Hamming subgraph, and it yields an efficient algorithm for finding (anti)median sets in Hamming graphs. We note that Algorithm 2 also extends to nonbipartite case, because the Hamming distance can be applied in partial Hamming graphs (we leave details to the reader).

In this paper we have embedded median graphs into hypercubes to obtain fast algorithms for computing median and antimedian sets. Since Theorem 3.1 is more general, it can perhaps be applied to other situations as well. In particular it could be useful to embed isometrically median graphs into other Cartesian products, say products of paths or trees. Moreover, one could embed isometrically other classes of graphs into appropriate host graphs, say ℓ_1-graphs.

Acknowledgment

We are grateful to the referee #3 for critical remarks that led to the more general approach (with a much simpler proof) of Theorem 3.1.

References

[1] K. Balakrishnan, *Algorithms for Median Computation in Median Graphs and Their Generalizations Using Consensus Strategies*, Ph.D Thesis, University of Kerala, 2006.

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

[3] H.-J. Bandelt and J.-P. Barthélemy, Medians in median graphs, Discrete Appl. Math. 8 (1984) 131–142.

[4] H.-J. Bandelt and V. Chepoi, Graphs with connected medians, SIAM J. Discrete Math. 15 (2002) 268–282.

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

[6] J.-P. Barthélemy and B. Monjardet, The median procedure in cluster analysis and social choice theory, Math. Social Sci. 1 (1980/81) 235–267.

[7] H. Bielak and M. M. Syslo, Peripheral vertices in graphs, Stud. Sci. Math. Hung. 18 (1983) 269–275.

[8] B. Brešar, W. Imrich and S. Klavžar, Fast recognition algorithms for classes of partial cubes, Discrete Appl. Math. 131 (2003) 51–61.

[9] P. Cappanera, G. Gallo and F. Maffioli, Discrete facility location and routing of obnoxious activities, Discrete Appl. Math. 133 (2003) 3–28.

[10] N. Chiba and T. Nishizeki, Arboricity and subgraph listing algorithms, SIAM J. Comput. 14 (1985) 210–223.

[11] T. Feder, Stable networks and product graphs, Mem. Amer. Math. Soc. 116 (1995) no. 555.

[12] J. Hagauer, W. Imrich and S. Klavžar, Recognizing median graphs in subquadratic time, Theoret. Comput. Sci. 215 (1999) 123–136.

[13] W. Imrich and S. Klavžar, Recognizing graphs of acyclic cubical complexes, Discrete Appl. Math. 95 (1999) 321–330.

[14] W. Imrich and S. Klavžar, *Product Graphs: Structure and Recognition*, Wiley-Interscience, New York, 2000.

[15] W. Imrich, S. Klavžar and H.M. Mulder, Median graphs and triangle-free graphs, SIAM J. Discrete Math. 12 (1999) 111–118.

[16] S. Klavžar and H. M. Mulder, Median graphs: characterizations, location theory and related structures, J. Combin. Math. Combin. Comput. 30 (1999) 103–127.

[17] B. Leclerc, The median procedure in the semilattice of orders, Discrete Appl. Math. 127 (2003) 285–302.

[18] F. R. McMorris, H. M. Mulder and F. R. Roberts, The median procedure on median graphs, Discrete Appl. Math. 84 (1998) 165–181.

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

[20] A. Tamir, Locating two obnoxious facilities using the weighted maximin criterion, Oper. Res. Lett. 34 (2006) 97–105.

[21] A. Taranenko and A. Vesel, Fast recognition of Fibonacci cubes, Algorithmica 49 (2007) 81–93.

[22] S. S. Ting, A linear-time algorithm for maxisum facility location on tree networks, Transportation Sci. 18 (1984) 76–84.

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

[24] B. Zmazek and J. Žerovnik, The obnoxious center problem on weighted cactus graphs, Discrete Appl. Math. 136 (2004) 377–386.