



HAL
open science

Diameter, radius and all eccentricities in linear time for constant-dimension median graphs

Pierre Bergé, Michel Habib

► **To cite this version:**

Pierre Bergé, Michel Habib. Diameter, radius and all eccentricities in linear time for constant-dimension median graphs. 2023. hal-04159957

HAL Id: hal-04159957

<https://hal.science/hal-04159957v1>

Preprint submitted on 12 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Diameter, radius and all eccentricities in linear time for constant-dimension median graphs*

Pierre Bergé[†] and Michel Habib

IRIF, CNRS, Université de Paris, France

May 27, 2021

Abstract

Median graphs form the class of graphs which is the most studied in metric graph theory. Recently, Bénéteau *et al.* [2019] designed a linear-time algorithm computing both the Θ -classes and the median set of median graphs. A natural question emerges: is there a linear-time algorithm computing the diameter and the radius for median graphs?

We answer positively to this question for median graphs G with constant dimension d , *i.e.* the dimension of the largest induced hypercube of G . We propose a combinatorial algorithm computing all eccentricities of median graphs with running time $O(2^{O(d \log d)} n)$. As a consequence, this provides us with a linear-time algorithm determining both the diameter and the radius of median graphs with $d = O(1)$, such as cube-free median graphs. As the hypercube of dimension 4 is not planar, it shows also that all eccentricities of planar median graphs can be computed in $O(n)$.

1 Introduction

We study one of the most fundamental problems in algorithmic graph theory related to distances: the *diameter* and the *radius*. Given an undirected graph $G = (V, E)$, the diameter is the maximum distance $d(u, v)$, $u, v \in V$, where $d(u, v)$ is the length of the shortest (u, v) -path. The eccentricity of a vertex v is the maximum length of a shortest path starting from v : the diameter is thus the maximum eccentricity and the radius is defined as the minimum eccentricity. Both the diameter and the radius are basic parameters used to apprehend the structure of a graph.

Multiple *Breadth First Search* (BFS) suffice as a naive algorithm to compute the distances between all pairs of vertices and, therefore, to obtain the diameter, the radius and all eccentricities in $O(n|E|)$ for a n -vertex graph. Unfortunately, no known algorithm is able to determine the diameter much faster than all distances. Furthermore, it was shown [1] that both computing the radius and all distances are equivalent problems under subcubic reductions. Very efficient algorithms have been proposed for the diameter on certain classes of graphs, for example [2, 18, 25]. Many works have also been devoted to approximation algorithms for this parameter. In particular, Chechik *et al.* [19] showed that the diameter can be approximated within a factor $\frac{3}{2}$ in time $O^*(m^{\frac{3}{2}})$. On sparse graphs, a natural question is whether we can compute exactly the diameter in subquadratic time. It was shown [38] that no $O(n^{2-\varepsilon})$ -time algorithm can achieve an approximation factor smaller than $\frac{3}{2}$ for the diameter on sparse graphs unless the Strong Exponential Time Hypothesis (SETH) fails.

Linear-time heuristics have been proposed to estimate the diameter in general graphs. The 2-SWEEP algorithm [33] consists in taking an arbitrary vertex r_1 of the graph, computing the

*An extended abstract of this paper will appear at LAGOS 2021. This is the full version.

[†]Email : berge@irif.fr

farthest-to- r_1 vertex a_1 with a BFS and eventually finding the farthest-to- a_1 vertex b_1 with another BFS. The distance $d(a_1, b_1)$ is returned. This method works well on some special classes of graphs. It was shown that 2-SWEEP returns exactly the diameter on trees [28]. Moreover, it approximates it with an additive error 1 on chordal graphs and an additive error 2 on AT-free graphs [23]. The 4-SWEEP algorithm [16] picks a vertex in the middle of a shortest (a_1, b_1) -path and computes another 2-SWEEP. It performs well on real-world undirected graphs [16].

In this paper, we propose a linear time algorithm computing the diameter, the radius, and all eccentricities of constant-dimension median graphs. Median graphs are the graphs such that any triplet of vertices has a unique median. Put formally, given $x, y, z \in V$, there is a unique vertex $m(x, y, z)$ lying at the same time on some shortest (x, y) , (y, z) , and (z, x) -paths. Said differently, $m(x, y, z)$ is the unique vertex being metrically between x and y , y and z , z and x . Median graphs are *partial cubes*, *i.e.* isometric subgraphs of hypercubes. However, partial cubes are not necessarily median. The dimension d of a median graph is the dimension of its largest induced hypercube. This parameter is upper-bounded by $\lfloor \log n \rfloor$.

Median graphs are related to numerous areas: universal algebra [4, 15], CAT(0) cube complexes [8, 20], abstract models of concurrency [13, 39], and genetics [11, 12]. They have strong structural properties and admit many characterizations, such as the Mulder's convex expansion [36, 37], and are related to hypercubes retracts [6], Cartesian products and gated amalgams [8], but also Helly hypergraphs [35]. Median graphs are bipartite and have at most $dn \leq n \log n$ edges. They do not contain induced $K_{2,3}$, otherwise a triplet of vertices would admit at least two medians. The cardinality function of hypercubes $\alpha_i(G)$ - the number of induced hypercubes of dimension $0 \leq i \leq d$ in the median graph G - verifies nice formulas [17, 32]. A key concept to understand the structure of median graphs is the equivalence relation Θ , which is the reflexive and transitive closure of relation Θ_0 , where two edges are in Θ_0 if they are opposite edges of a common 4-cycle. An equivalence class of Θ is called a Θ -class. Each Θ -class of a median graph form a matching cutset, splitting the graph into two connected components, called halfspaces. These halfspaces are convex. The number q of Θ -classes is less than n and, moreover, it is exactly the dimension of the hypercube in which the median graph G isometrically embeds. It was shown in [31] that value q satisfies the Euler-type formula $2n - m - q \leq 2$. The Θ -classes can be identified in linear time $O(|E|) = O(dn)$, using a Lexicographic BFS [14].

To the best of our knowledge, there is no subquadratic algorithm for the diameter on median graphs. Bénéteau *et al.* [14] and Ducoffe [24] recently formulated this open question. There exist efficient algorithms for other metric parameters on median graphs. Thanks to the linear time computation of Θ -classes, the median set and the Wiener index can be determined in $O(|E|)$ [14]. The median set, *i.e.* the vertices u of G which minimize $\sum_{v \in V} d(u, v)$, satisfies the majority rule [5, 7]. Another challenging question is the recognition of median graphs. Two very efficient algorithms have been proposed. Using convex characterizations of halfspaces, Hagauer *et al.* [26] showed that median graphs can be recognized in $O(n^{\frac{3}{2}}\sqrt{n})$. In [29], a bijection between median graphs and triangle-free graphs is identified. As a consequence, the recognition times of these two families are the same, if we neglect poly-logarithmic factors. Taking a very efficient algorithm which detects triangle-free graphs [3] produces a recognition of median graphs running in $O((n \log^2 n)^{1.41})$.

Subfamilies of median graphs have also been studied in the literature. There is an algorithm computing the diameter and the radius in linear time for squaregraphs, which are planar cube-free median graphs where all inner vertices have degree at least four [10, 21]. Recently, distance and routing labeling schemes [22] of size $O(\log^3 n)$ were designed for cube-free median graphs, *i.e.* median graphs verifying $d \leq 2$.

In this paper, we construct an algorithm computing the diameter for median graphs which runs in time $O(n)$ when $d = O(1)$. Concretely, our algorithm returns the exact value of the diameter for all median graphs G in time $O(f(d)n)$, where $f(d)$ is an exponential fonction of d . It can be naturally extended to provide us with all eccentricities of the median graph with the same running time. Our contribution is summarized below.

Main result. *There is a combinatorial algorithm which computes the diameter, the radius and all eccentricities in time $O^*(2^{d(\log(d)+1)}n)$ on median graphs.*

Notation O^* neglects polynomials of d , which are also poly-logarithmic factors of n . A consequence of our main result is that, for any $d = O(1)$, all eccentricities can be determined in linear time $O(n)$ on Q_d -free median graphs. For example, this is the case for cube-free median graphs

($d \leq 2$). Moreover, as Q_4 is not planar, planar median graphs are Q_4 -free, so our algorithm is linear for this family of graphs. Obviously, such an algorithm is not linear for all median graphs, as parameter d can attain value $\log n$.

Given that 2-SWEEP returns in linear time the diameter on trees, which are the median graphs of dimension $d = 1$, one can wonder whether the heuristics 2-SWEEP and 4-SWEEP can compute the diameter of median graphs for larger values of d . Another contribution in this paper is the proof that it is not the case. We provide median graphs of dimension 2 on which 2-SWEEP and 4-SWEEP fail to determine the diameter.

The organization of the paper follows. In Section 2, we introduce the definitions and notions related to median graphs. We describe in detail the properties of Θ -classes, in particular the orthogonality, which is a key concept in this work. In Section 3, we show that 2-SWEEP and 4-SWEEP do not allow us to determine the exact value of the diameter on median graphs. We provide instances on which they are unsuccessful. In Section 4, we present a first version of our algorithm which computes the diameter only. We proceed in two steps: the computation of labels for the hypercubes of G (Section 4.1) and the reduction to an optimization problem we call MAXIMUM-WEIGHTED DISJOINT SETS (Section 4.2). In Section 5, we introduce an additional step in our algorithm to determine the eccentricity of each vertex of G . We conclude this work in Section 6 by listing possible lines of future research on median graphs.

2 Preliminaries

In this section, we recall some basic notions related to graph theory, distances, and more particularly median graphs. The results listed are either reminded from the literature or direct consequences of earlier works.

2.1 Graphs and distances

All graphs $G = (V, E)$ considered in this paper are undirected, unweighted, simple, finite and connected. Edges $(u, v) \in E$ are sometimes denoted by uv to improve the readability of our article. Given two vertices $u, v \in V$, we denote by $d(u, v)$ the *distance* between u and v , *i.e.* the length of the shortest (u, v) -path. The interval $I(u, v)$ contains exactly the vertices which are on shortest (u, v) -paths. Put formally,

$$I(u, v) = \{x \in V : d(u, x) + d(x, v) = d(u, v)\}$$

We denote by $\text{ecc}(u)$ the *eccentricity* of a vertex $u \in V$, *i.e.* the maximum distance $d(u, v)$ for all $v \in V$: $\text{ecc}(u) = \max_{v \in V} d(u, v)$. The diameter $\text{diam}(G)$ of graph G is its maximum eccentricity: $\text{diam}(G) = \max_{u \in V} \text{ecc}(u)$. Concretely, the diameter is the length of the longest shortest path of G . The radius $\text{rad}(G)$ is the minimum eccentricity: $\text{rad}(G) = \min_{u \in V} \text{ecc}(u)$.

Let $H \subseteq V$ be a vertex set. We say that H (or the induced subgraph $G[H]$) is *convex* if $I(u, v) \subseteq H$ for any pair $u, v \in H$. We say that H is *gated* if any vertex $v \notin H$ admits a gate $v' \in H$, *i.e.* a vertex that belongs to all intervals $I(v, x)$, $x \in H$. In other words, for any $x \in H$, we have $d(v, v') + d(v', x) = d(v, x)$.

Given an integer $k \geq 1$, the k -dimensional hypercube Q_k represents all the subsets of $\{1, \dots, k\}$ as the vertex set. Two subsets A and B are connected by an edge if they differ only by one element, *i.e.* $|A \Delta B| = 1$. The hypercube Q_2 is called a *square*.

2.2 Median graphs

A graph is *median* if, for any triplet x, y, z of distinct vertices, the set $I(x, y) \cap I(y, z) \cap I(z, x)$ contains exactly one vertex called the median $m(x, y, z)$. A median graph is bipartite, triangle-free, and does not contain an induced $K_{2,3}$ [8, 27, 36]. Some well-known families of graphs are median: trees, grids, squaregraphs [10], and hypercubes. The dimension $d = \text{dim}(G)$ of a median graph G is the dimension of the largest hypercube contained in G as an induced subgraph.

Figure 1 presents three examples of median graphs. One has dimension $d = 1$ and is (necessarily) a tree. One has dimension $d = 2$, a cogwheel. Cogwheels are squaregraphs but certain median

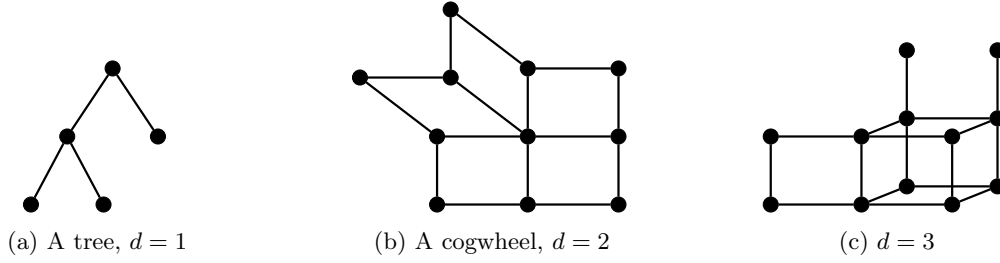


Figure 1: Examples of median graphs

graphs of dimension 2 are not squaregraphs [10]. The last one has dimension $d = 3$ as it contains an hypercube Q_3 .

We provide a list of properties satisfied by median graphs. In particular, we define the notion of Θ -classes which is a key ingredient of several existing algorithms [14, 26, 29].

In general graphs, all gated subgraphs are convex. The reverse is true in median graphs.

Lemma 1 (Convex \Leftrightarrow Gated [8, 14]). *Any convex subgraph of a median graph is gated.*

We remind the notion of Θ -class, which is well explained in [14], and enumerate some properties related to it. We say that the edges uv and xy are in relation Θ_0 if they form a square $uvyx$, where uv and xy are opposite edges. Then, Θ refers to the reflexive and transitive closure of relation Θ_0 . The classes of the equivalence relation Θ are denoted by E_1, \dots, E_q . We denote by \mathcal{E} the set of Θ -classes: $\mathcal{E} = \{E_1, \dots, E_q\}$. Parameter q is less than the number of vertices n . To avoid confusions, let us highlight that parameter q is different from the dimension d : for example, on trees, $d = 1$ whereas $q = n - 1$. Moreover, the dimension d is at most $\lfloor \log n \rfloor$.

Lemma 2 (Θ -classes in linear time [14]). *There is an algorithm which computes the Θ -classes E_1, \dots, E_q of a median graph in linear time $O(|E|)$.*

In median graphs, each class E_i , $1 \leq i \leq q$, is a perfect matching cutset and its two sides H'_i and H''_i verify nice properties, that are presented below.

Lemma 3 (Halfspaces of E_i [14, 26, 37]). *For any $1 \leq i \leq q$, the graph G deprived of edges of E_i , i.e. $G \setminus E_i = (V, E \setminus E_i)$, has two connected components H'_i and H''_i , called halfspaces. Edges of E_i form a matching: they have no endpoint in common. Halfspaces satisfy the following properties:*

- Both H'_i and H''_i are convex/gated.
- If uv is an edge of E_i with $u \in H'_i$ and $v \in H''_i$, then $H'_i = W(u, v) = \{x \in V : d(x, u) < d(x, v)\}$ and $H''_i = W(v, u) = \{x \in V : d(x, v) < d(x, u)\}$.

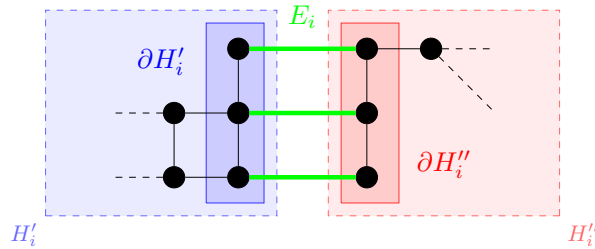


Figure 2: A class E_i with sets $H'_i, H''_i, \partial H'_i, \partial H''_i$

We denote by $\partial H'_i$ the subset of H'_i containing the vertices which are adjacent to a vertex in H''_i . Put differently, set $\partial H'_i$ is made up of vertices of H'_i which are endpoints of edges in E_i . Similarly, set $\partial H''_i$ contains the vertices of H''_i which are adjacent to H'_i . We say these sets are the *boundaries* of halfspaces H'_i and H''_i respectively.

Lemma 4 (Boundaries [14, 26, 37]). *Both $\partial H'_i$ and $\partial H''_i$ are convex/gated. Moreover, the edges of E_i define an isomorphism between $\partial H'_i$ and $\partial H''_i$.*

As a consequence, suppose uv and $u'v'$ belong to E_i : if uu' is an edge and belongs to class E_j , then vv' is an edge of E_j too. We terminate this list of lemmas with a last property dealing with the orientation of edges from a canonical basepoint $v_0 \in V$. The v_0 -orientation of the edges of G according to v_0 is such that, for any edge uv , the orientation is \vec{uv} if $d(v_0, u) < d(v_0, v)$. Indeed, we cannot have $d(v_0, u) = d(v_0, v)$ as G is bipartite.

Lemma 5 (Orientation [14]). *All edges can be oriented according to any canonical basepoint v_0 .*

From now on, we suppose that an arbitrary basepoint $v_0 \in V$ has been selected and we refer automatically to the v_0 -orientation when we mention incoming or outgoing edges.

2.3 Orthogonal Θ -classes and hypercubes

We present now an important notion on median graphs. In [32], Kovse studied a relationship between *splits* which refer to the halfspaces of Θ -classes. It says that two splits $\{H'_i, H''_i\}$ and $\{H'_j, H''_j\}$ are *incompatible* if the four sets $H'_i \cap H'_j$, $H'_i \cap H''_j$, $H''_i \cap H'_j$, and $H''_i \cap H''_j$ are nonempty. We use different notation to characterize this relationship which makes more sense in our context.

Definition 1 (Orthogonal classes). *We say that classes E_i and E_j are orthogonal ($E_i \perp E_j$) if there is a square $uvyx$ in G , where $uv, xy \in E_i$ and $ux, vy \in E_j$.*

We observe that classes E_i and E_j are orthogonal if and only if the splits produced by their halfspaces are incompatible.

Lemma 6 (Orthogonal \Leftrightarrow Incompatible). *E_i and E_j are orthogonal if and only if $\{H'_i, H''_i\}$ and $\{H'_j, H''_j\}$ are incompatible.*

Proof. First, if E_i and E_j are orthogonal, there is a square containing the edges of these two classes. The four vertices belong respectively to the four sets $H'_i \cap H'_j$, $H'_i \cap H''_j$, $H''_i \cap H'_j$, and $H''_i \cap H''_j$. Consequently, these sets are nonempty and the splits are incompatible.

Second, suppose the four sets $H'_i \cap H'_j$, $H'_i \cap H''_j$, $H''_i \cap H'_j$, and $H''_i \cap H''_j$ are nonempty. Consider only the vertices in H'_i . As $H'_i \cap H'_j$ and $H'_i \cap H''_j$ are nonempty, we take arbitrarily one vertex x_1 in $H'_i \cap H'_j$ and one vertex x_2 in $H'_i \cap H''_j$. By convexity of H'_i (Lemma 3), any shortest (x_1, x_2) -path passes through an edge of E_j with its two endpoints in H'_i : this edge is denoted by y_1y_2 , where $y_1 \in H'_i \cap \partial H'_j$ and $y_2 \in H'_i \cap \partial H''_j$. If we consider now only the vertices in H''_i , we can point out an edge y_3y_4 with the same method, where $y_3 \in H''_i \cap \partial H'_j$ and $y_4 \in H''_i \cap \partial H''_j$. By convexity of $\partial H'_j$ (Lemma 4), any shortest (y_1, y_3) -path passes through an edge of E_i with two endpoints in $\partial H'_j$: we denote this edge by z_1z_3 , where $z_1 \in \partial H'_i \cap \partial H'_j$ and $z_3 \in \partial H''_i \cap \partial H'_j$. Let z_2 (resp. z_4) be the neighbor of z_1 (resp. z_3) in $\partial H''_j$. We have $z_1z_2, z_3z_4 \in E_j$ and $z_1z_3 \in E_i$. As E_j defines an isomorphism (Lemma 4) between $\partial H'_j$ and $\partial H''_j$, then $z_2z_4 \in E$. We obtain a square $z_1z_2z_4z_3$ with edges belonging to classes E_i and E_j . \square

This results proves implicitly that the splits $\{H'_i, H''_i\}$ and $\{H'_j, H''_j\}$ are incompatible if and only if the following four sets formed with boundaries $\partial H'_i \cap \partial H'_j$, $\partial H''_i \cap \partial H'_j$, $\partial H'_i \cap \partial H''_j$, and $\partial H''_i \cap \partial H''_j$ are nonempty.

We pursue with a property on orthogonal classes: if two edges of two orthogonal classes E_i and E_j are incident, they belong to a common square. Even if the result was already proposed in [13], we present a different proof consistent with the notions evoked earlier.

Lemma 7 (Squares [13]). *Let $xu \in E_i$ and $uy \in E_j$. If E_i and E_j are orthogonal, then there is a vertex v such that $uyvx$ is a square.*

Proof. We say w.l.o.g. that $u \in H'_i \cap H'_j$. Then, $x \in H''_i \cap H'_j$ and $y \in H'_i \cap H''_j$. Let z be an arbitrary vertex of $H''_i \cap H''_j$ which is nonempty (Lemma 6). We fix $v = m(x, y, z)$. We prove that vertex v is different from x , u , and y .

As H''_i is convex, $I(x, z) \subseteq H''_i$ and $v \in H''_i$. Consequently, $v \neq u, y$. Similarly, as H'_j is convex, $I(y, z) \subseteq H'_j$, so $v \neq x$. The median $v = m(x, y, z)$ belongs to $I(x, y)$ and the distance $d(x, y)$ is at most 2. As $v \neq x, u, y$, vertex v belongs to a shortest (x, y) -path of length 2 which does not contain u . As a consequence, v is adjacent to x and y . \square

Now, we focus on set of classes which are pairwise orthogonal.

Definition 2 (Pairwise Orthogonal Family). *We say that a set of classes $X \subseteq \mathcal{E}$ is a Pairwise Orthogonal Family (POF) if for any pair $E_j, E_h \in X$, we have $E_j \perp E_h$.*

The empty set is considered as a POF. The notion of POF is strongly related to the induced hypercubes in median graphs. First, observe that all Θ -classes of a median graph form a POF if and only if the graph is an hypercube of dimension $\log n$ [32, 34]. Second, if all classes of a POF X are adjacent to $v \in V$, there is an hypercube containing v and its edges belong to classes in X .

Lemma 8 (POFs adjacent to a vertex). *Let X be a POF, $v \in V$, and assume that for each class $E_i \in X$, there is an edge of E_i adjacent to v . There exists an hypercube Q containing vertex v and all edges of X adjacent to v . Moreover, the Θ -classes of the edges of Q are the classes of X .*

Proof. We proceed inductively. If $|X| = 1$, $X = \{E_i\}$, then the edge e_i adjacent to v is an hypercube of dimension 1 and v is one of its endpoints. If $|X| = 2$, $X = \{E_i, E_j\}$, Lemma 7 shows that there is a square containing v and the two edges of X adjacent to v .

Assume that for a POF X adjacent to v , $|X| \leq k$, there is an hypercube Q containing v and such that its edges belong to the classes in X . Consider now a POF X , $|X| = k + 1$, say w.l.o.g. $X = \{E_1, \dots, E_{k+1}\}$, adjacent to v . We denote by e_1, \dots, e_{k+1} the edges belonging respectively to classes E_1, \dots, E_{k+1} which are adjacent to v . Indeed, we cannot have two edges of the same class E_i adjacent to v as any E_i is a matching (Lemma 3). Let $X' = \{E_1, \dots, E_k\}$ and Q' be the hypercube containing v and admitting edges in X' . We say w.l.o.g. that vertex v belongs to the boundary $\partial H'_{k+1}$ of class E_{k+1} .

We apply Lemma 7 to all pairs of edges (e_i, e_{k+1}) with $1 \leq i \leq k$. They have a common endpoint v . All these pairs belong in fact to a square of classes $\{E_i, E_{k+1}\}$. As a consequence, all vertices in Q' which are at distance 1 from v belong to $\partial H'_{k+1}$. We can pursue this reasoning for the vertices of Q' at distance 2 from v : the edges of Q' connecting vertices at distance 1 with vertices at distance 2 from v belong to X' . Moreover, the vertices at distance 1 are endpoints of edges in E_{k+1} . According to Lemma 7, the vertices of Q' at distance 2 from v belong to $\partial H'_{k+1}$. Finally, after multiple applications of Lemma 7, we prove that all vertices in Q' belong to $\partial H'_{k+1}$.

As class E_{k+1} forms an isomorphism between $\partial H'_{k+1}$ and $\partial H''_{k+1}$, the matching E_{k+1} connects a hypercube Q' with a isomorphic hypercube Q'' in $\partial H''_{k+1}$. The induced subgraph on Q' and Q'' is the Cartesian product between Q_k and K_2 : it forms an hypercube Q of dimension $k + 1$. As $v \in Q'$, then $v \in Q$. Moreover, the Θ -classes represented in Q are exactly $X' \cup \{E_{k+1}\} = X$. \square

In fact, there is a bijection between the POFs and the vertices of a median graph. Given a POF X , the vertex which is associated with it is the farthest-to- v_0 vertex of the closest-to- v_0 hypercube formed with Θ -classes X .

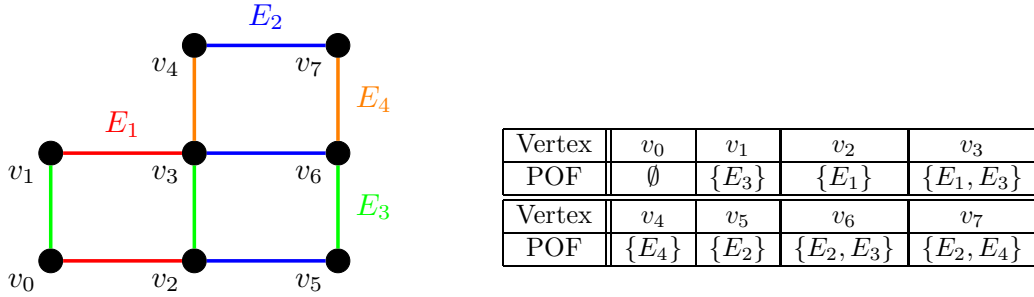


Figure 3: Illustration of the bijection between V and the set of POFs.

Lemma 9 (POFs and hypercubes [9, 11, 32]). *Consider an arbitrary canonical basepoint $v_0 \in V$ and the v_0 -orientation for the median graph G . Given a vertex $v \in V$, let $N^-(v)$ be the set of edges going into v according to the v_0 -orientation. Let $\mathcal{E}^-(v)$ be the classes of the edges in $N^-(v)$. The following propositions are true:*

- For any vertex $v \in V$, $\mathcal{E}^-(v)$ is a POF. Moreover, vertex v and the edges of $N^-(v)$ belong to an induced hypercube formed by the classes $\mathcal{E}^-(v)$.
- For any POF X , there is a unique vertex v_X such that $\mathcal{E}^-(v_X) = X$. Vertex v_X is the closest-to- v_0 vertex v such that $X \subseteq \mathcal{E}^-(v)$.
- The number of POFs in G is equal to the number n of vertices.

This result highlights a bijection between the vertices V and the POFs of G . An example is given in Figure 3 with a graph of dimension $d = 2$. Edges are colored in function of their Θ -class. Vertex v_0 is the canonical basepoint. For example, $v_1v_3 \in E_1$. We associate with any POF X of G the vertex v satisfying $\mathcal{E}^-(v) = X$ with the v_0 -orientation. Obviously, the empty POF is associated with v_0 which has no incoming edges.

This bijection can be used to enumerate the POFs of a median graph in linear time [11, 32]. Given a basepoint v_0 , we say that the *basis* (resp. *anti-basis*) of an induced hypercube Q is the single vertex v such that all edges of the hypercube adjacent to v are outgoing (resp. ingoing) from v . Said differently, the basis of Q is its closest-to- v_0 vertex and its anti-basis is its farthest-to- v_0 vertex. What Lemma 9 states is also that we can associate with any POF X an hypercube Q_X which contains exactly the classes X and admits v_X as its anti-basis. Moreover, the hypercube Q_X is the closest-to- v_0 hypercube formed with the classes in X . Figure 4 shows a vertex v with its ingoing and outgoing edges with the v_0 -orientation. The dashed edges represent the hypercube with anti-basis v and POF $\mathcal{E}^-(v)$.

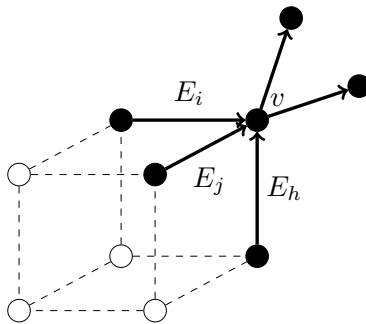


Figure 4: Edges ingoing to a vertex v belong to an hypercube made up of their classes $\mathcal{E}^-(v) = \{E_i, E_j, E_h\}$ and with anti-basis v .

This observation implies that the number of POFs is less than the number of hypercubes in G . We remind a formula establishing a relationship between the number of POFs and the number of hypercubes in the literature. Let $\alpha(G)$ (resp. $\beta(G)$) be the number of hypercubes (resp. POFs) in G . Let $\beta_i(G)$ be the number of POFs of cardinality $i \leq d$ in G . According to [11, 32], we have:

$$\alpha(G) = \sum_{i=0}^d 2^i \beta_i(G) \quad (1)$$

Equation (1) produces a natural upper bound for the number of hypercubes.

Lemma 10 (Number of hypercubes). $\alpha(G) \leq 2^d n$.

Proof. This is the consequence of $\beta(G) = \sum_{i=0}^d \beta_i(G) = n$ (Lemma 9). \square

Each hypercube in the median graph G can be defined with only its anti-basis v and the edges \widehat{N} of the hypercube that are adjacent and going into v according the v_0 -orientation. These edges are a subset of $N^-(v)$: $\widehat{N} \subseteq N^-(v)$. Conversely, given a vertex v , each subset of $N^-(v)$ produces an hypercube which admits v as an anti-basis (this hypercube is a sub-hypercube of the one obtained with v and $N^-(v)$, Lemma 9). Another possible bijection is to consider an hypercube as a pair composed of its anti-basis v and the Θ -classes $\widehat{\mathcal{E}}$ of the edges in \widehat{N} .

From the previous results and Lemma 10, we deduce an algorithm enumerating the hypercubes in G in time $O(d2^d n)$.

Lemma 11 (Enumeration of hypercubes). *We can enumerate all triplets $(v, u, \hat{\mathcal{E}})$, where v is the anti-basis of an hypercube Q , u its basis, and $\hat{\mathcal{E}}$ the Θ -classes of the edges of Q in time $O(d2^d n)$. Moreover, the list obtained fulfils the following partial order: if $d(v_0, v) < d(v_0, v')$, then any triplet $(v, u, \hat{\mathcal{E}})$ containing v appears before any triplet $(v', u', \hat{\mathcal{E}}')$ containing v' .*

Proof. We run a BFS from a canonical v_0 to obtain the v_0 -orientation. For any vertex v visited, we list all subsets of $N^-(v)$. Each of these subsets \hat{N} correspond to an hypercube with anti-basis v . For each \hat{N} , we determine the Θ -classes $\hat{\mathcal{E}}$ of its edges. For any pair $(v, \hat{\mathcal{E}})$, we compute the basis of this hypercube: we start a walk from v and we traverse once an edge for each class in $\hat{\mathcal{E}}$ in any order. The vertex we visit at the end of the walk is the basis u . At the end of the execution, we have a list of triplets $(v, u, \hat{\mathcal{E}})$, where $\hat{\mathcal{E}} \subseteq \mathcal{E}^-(v)$. The execution time is equal the number of hypercubes multiplied by the size of the triplets representing them, which is upper-bounded by $d + 1$. The partial order evoked in the statement of this lemma is due to the BFS which visits v before v' if $d(v_0, v) < d(v_0, v')$. \square

The enumeration of hypercubes is executed in linear time for median graphs with $d = O(1)$. It will be used in the remainder to describe our algorithm computing the diameter.

3 Failure of BFS-based heuristics on median graphs

We prove in this section that 2-SWEEP and 4-SWEEP, two well-known linear time heuristics for the diameter in general graphs, do not determine the exact value of diameter in median graphs. We begin with a short introduction of these algorithms and then present two median graphs on which they are unsuccessful.

The heuristic 2-SWEEP consists in two successive BFS returning a distance $d(a_1, b_1)$ between two vertices a_1 and b_1 which is supposed to estimate the diameter of the graph. First, it starts from a random vertex of G denoted by r_1 . It computes a first BFS starting from r_1 to determine the farthest-to- r_1 vertex a_1 . Formally, vertex a_1 verifies $d(r_1, a_1) = \max_{v \in V} d(r_1, v)$. Then, it computes a second BFS, starting from a_1 to determine the farthest-to- a_1 vertex b_1 . Vertex b_1 verifies $d(a_1, b_1) = \max_{v \in V} d(a_1, v)$. Value $d(a_1, b_1)$ is returned.

The heuristic 4-SWEEP consists in four successive BFS. We start with a first 2-SWEEP which enables us to obtain vertices a_1 and b_1 . Then, we determine vertex r_2 which is the middle of a shortest (a_1, b_1) -path. We compute a second 2-SWEEP starting from vertex r_2 which gives vertices a_2 and b_2 . Value $d(a_2, b_2)$ is returned.

Unfortunately, both of these fast and simple algorithms fail to determine the diameter on median graphs.

Theorem 1. *There are two median graphs G^* and H^* of dimension 2 such that 2-SWEEP does not return the diameter on G^* and 4-SWEEP does not return it on H^* .*

Proof. We begin with graph G^* and the execution of 2-SWEEP. Graph G^* is a square with a pendant vertex. Figure 5a represents G^* and indicates the identity of r_1 , a_1 , and b_1 . We start from the arbitrary vertex r_1 . Its eccentricity is $\text{ecc}(r_1) = 2$. Vertex a_1 verifies $d(r_1, a_1) = 2$ and is

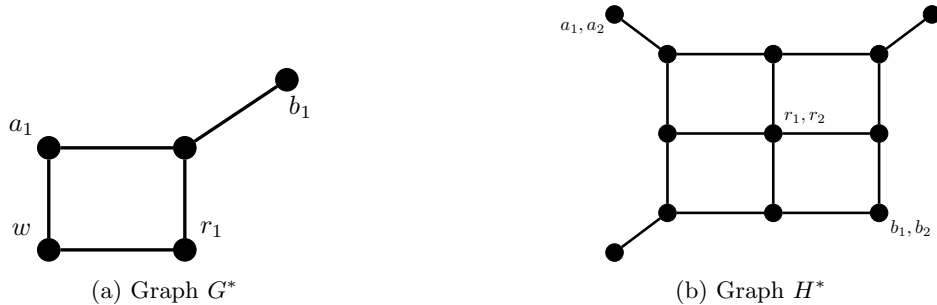


Figure 5: Different steps of 2-SWEEP and 4-SWEEP on graphs G^* and H^* .

selected as the start of the next BFS. Similarly, $\text{ecc}(a_1) = 2$ and b_1 is selected as the farthest-to- a_1 vertex. The distance returned is $d(a_1, b_1) = 2$ whereas $\text{diam}(G^*) = d(b_1, w) = 3$.

We focus now on graph H^* and a possible execution of 4-SWEEP. Graph H^* is a (3×3) -grid with pendant vertices. Figure 5b represents H^* and reveals the identity of r_1 , a_1 , b_1 , a_2 and b_2 . We start at vertex r_1 . Vertex a_1 maximizes $d(r_1, v)$ and b_1 maximizes $d(a_1, v)$. Then, vertex r_2 is a middle of a shortest (a_1, b_1) -path, but $r_2 = r_1$. Thus, pair a_1 and b_1 can be selected as a potential diameter again and again. We have $a_1 = a_2$ and $b_1 = b_2$. In this case, the distance returned is $d(a_2, b_2) = 5$ while $\text{diam}(G) = 6$. \square

This result convinces us that a more involved algorithm is needed if we aim at determining not only the diameter of constant-dimension median graphs but also all its eccentricities.

4 Computing the diameter in linear time for dimension $d = O(1)$

We proceed in two steps. First, we compute labels for each hypercube of G : they characterize the shortest paths of G starting at the basis of the hypercube and passing through its anti-basis (Section 4.1). Second, thanks to these labels, we reduce the diameter problem to MAXIMUM-WEIGHTED DISJOINT SETS (Section 4.2). We design an algorithm with running time $O^*(2^{d(\log(d)+1)}n)$ which identifies the diameter and a diametral pair for median graphs. It will be extended to obtain all eccentricities with the same execution time in Section 5.

4.1 Labels on hypercubes

A naive approach consists, for each vertex $v \in V$, in computing a BFS to determine its distance to all other vertices of G . In this way, we associate with any vertex v a label $\varphi(v)$ of size $O(n)$ indicating all these distances. Such a method produces necessarily a quadratic running time.

To determine the diameter of constant-dimension median graphs in linear time, we propose different labels which contain less information but enough for our problem. We label all pairs which contain each a vertex and a POF outgoing from this vertex. We remind that these pairs are in bijection with the set of hypercubes, according to Lemma 9: the vertex is the basis of the hypercube and the POFs contains exactly the Θ -classes of the hypercube. The idea is that the POFs indicate some “direction” in which we can go from the vertex. The label provides us with the distance of the longest shortest path we can find starting from the input vertex (the basis of the hypercube) and passing through the anti-basis of the hypercube. We will give more intuition on these labels when certain notions will be presented formally in this section.

As announced above, we fix an arbitrary canonical basepoint v_0 and for each class E_i , we say that the halfspace containing v_0 is H'_i .

Signature. Given two vertices $u, v \in V$, we define the set which contains the Θ -classes separating u from v .

Definition 3 (Signature $\sigma_{u,v}$). *We say that the signature of the pair of vertices u, v , denoted by $\sigma_{u,v}$, is the set of classes E_i such that u and v are separated in $G \setminus E_i$. In other words, u and v are in different halfspaces of E_i .*

All shortest (u, v) -paths contain exactly one edge for each class in $\sigma_{u,v}$.

Theorem 2. *For any shortest (u, v) -path P , the edges in P belong to classes in $\sigma_{u,v}$ and, for any $E_i \in \sigma_{u,v}$, there is exactly one edge of E_i in path P .*

Proof. Each Θ -class in $\sigma_{u,v}$ separates u from v . Consequently, a shortest (u, v) -path necessarily passes through an edge of each of these classes. Let $E_i \in \sigma_{u,v}$. A shortest (u, v) -path passes through E_i only once, otherwise we would come back to the first halfspace containing u (say H'_i w.l.o.g.) which is a contradiction with the convexity of H'_i (Lemma 3). Finally, suppose there is a class $E_j \notin \sigma_{u,v}$ represented in a shortest (u, v) -path. Vertices u and v are inside the same halfspace of E_j which is a contradiction with the convexity of halfspaces again: the shortest (u, v) -path cannot traverse an edge of E_j . \square

The converse also holds: if a path contains at most one edge of each class, it is a shortest path.

Theorem 3. *A path containing at most one edge of each Θ -class is a shortest path between its departure and its arrival.*

Proof. We proceed by induction on the length k of the path. If $k = 1$, the path consists in one edge only, so it is obviously a shortest path. Suppose we have a path P of length $k + 1$ and that all paths of length k having at most one edge of each class are shortest paths. Let u denote the departure, v the arrival and w the vertex before the arrival. We know by induction that the section $P^{(u,w)}$ of the path P between u and w is a shortest (u, w) -path. Let E_j denote the Θ -class of edge wv . There is no edge of class E_j in path $P^{(u,w)}$, so all vertices in $P^{(u,w)}$ are in the same halfspace of E_j , say H'_j w.l.o.g.. Vertex w is the gate of v in H'_j because $d(v, w) = 1$: no other vertex can be the gate of v , otherwise $d(v, w) \geq 2$. As $w \in I(u, v)$, path P is a shortest (u, v) -path. \square

The concept of signature can be generalized: given a set of edges, its signature is the set of Θ -classes represented in that set. For example, the signature of a path is the set of classes which have at least one edge in this path. In this way, the signature $\sigma_{u,v}$ is also the signature of any shortest (u, v) -path. The signature of a hypercube is the set of Θ -classes represented in its edges.

Ladder sets. Our idea is now to use POFs to characterize, given two vertices $u, v \in V$, the “direction” in which the shortest (u, v) -paths are oriented. We define the *ladder set* $L_{u,v}$, a subset of $\sigma_{u,v}$ which provides the hypercube you have to go through if you want to reach v from u with a shortest path. We see u as the departure vertex, while v is the arrival. The notion of ladder set is defined only for vertices u, v satisfying $u \in I(v_0, v)$. Another characterization involving two ladder sets will follow for any pair of vertices.

Definition 4 (Ladder set $L_{u,v}$). *Let $u, v \in V$ such that $u \in I(v_0, v)$. The ladder set $L_{u,v}$ is the subset of $\sigma_{u,v}$ which contains the classes admitting an edge adjacent to u .*

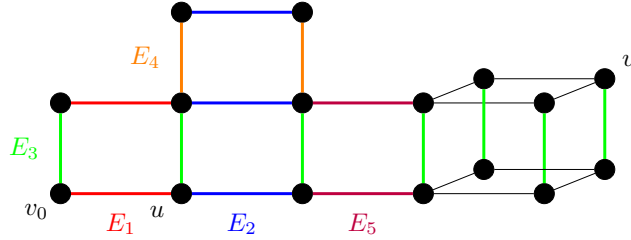


Figure 6: An example with $u \in I(v_0, v)$ and the ladder set $L_{u,v} = \{E_2, E_3\}$

Observe that when $u \in I(v_0, v)$, all edges of any shortest (u, v) -path are oriented towards v with the v_0 -orientation. The definition of the ladder set takes into account an order of the pair (u, v) while it is not the case for the signature. Figure 6 shows a median graph G and a pair u, v of vertices satisfying $u \in I(v_0, v)$. We color certain Θ -classes of this graph. The Θ -classes belonging to $\sigma_{u,v}$ and which are adjacent to u are E_2 and E_3 : $L_{u,v} = \{E_2, E_3\}$. For example, $E_5 \in \sigma_{u,v}$ but there is no edge of E_5 adjacent to u . We show that any ladder set is a POF.

Theorem 4. *Suppose $u \in I(v_0, v)$. Any ladder set $L_{u,v}$ is a POF. Moreover, there exists a shortest (u, v) -path, where its first edges belong to $L_{u,v}$.*

Proof. If $|L_{u,v}| = 1$, then the proof is terminated. Suppose $|L_{u,v}| \geq 2$. We denote by E_i and E_j two arbitrary classes in $L_{u,v}$. We prove that they are orthogonal.

Let $e_i = (u, z_i)$ be the edge of E_i adjacent to u . The one of class E_j is denoted by $e_j = (u, z_j)$. As z_i is the gate of u in H'_i , we have $d(z_i, v) = d(u, v) - 1$. Moreover, $\sigma_{z_i, v} = \sigma_{u, v} \setminus \{E_i\}$. Any shortest (z_i, v) -path concatenated with edge (u, z_i) produces a shortest (u, v) -path. Furthermore, it contains an edge of E_j , so one of its vertex is in $\partial H'_j$. In summary, we know that set $\partial H'_j$ is convex (Lemma 4) and contains both u and a vertex of any shortest (z_i, v) -path. Consequently, $z_i \in \partial H'_j$: there is an edge $(z_i, w) \in E_j$. As the edges of E_j define an isomorphism (Lemma 4), $(w, z_j) \in E$. We obtain a square made up of classes E_i and E_j , so $E_i \perp E_j$. As it is true for two arbitrary classes $E_i, E_j \in L_{u,v}$, set $L_{u,v}$ is a POF.

According to Lemma 8, there is an hypercube Q containing u and with signature $L_{u,v}$. As $u \in \partial H'_i$ for any $E_i \in L_{u,v}$, vertex u is the basis of hypercube Q . Let u^+ be the opposite vertex of u in Q . Consider the concatenation of a shortest (u, u^+) -path P made up of edges in Q and of a shortest (u^+, v) -path P^+ . The signature of P is $L_{u,v}$. As $L_{u,v} \subseteq \sigma_{u,v}$, vertex u^+ is in $I(u, v)$. Consequently, the concatenation $P \cdot P^+$ is a shortest (u, v) -path and its first edges belong to the Θ -classes in $L_{u,v}$. \square

More generally, for any ordering τ of the classes of $L_{u,v}$, we can identify a shortest (u, v) -path, where its first part is made up with edges of the classes of $L_{u,v}$ following the ordering τ . This prefix is in fact a shortest path between u and its opposite vertex in the hypercube containing u with signature $L_{u,v}$. However, several shortest (u, v) -paths may contain classes of $\sigma_{u,v} \setminus L_{u,v}$ before certain classes of $L_{u,v}$.

The main principle of our algorithm is to label each pair made up of a vertex $u \in V$ and a POF L “outgoing” from u , *i.e.* such that, for each $E_i \in L$, there is an edge of E_i outgoing from u . We know from Lemma 8 that it means there is an hypercube made up of edges in L and with basis u . Then, the label associated with u and L is the maximum distance $d(u, v)$ for a pair (u, v) , $u \in I(v_0, v)$, admitting L as its ladder set. There is one label for each hypercube of G .

We showed how the ladder sets characterize pairs of vertices u, v , where u, v , and v_0 are aligned, *i.e.* $u \in I(v_0, v)$. We focus now on the general case: we can characterize any pair u, v of vertices with two ladder sets.

Definition 5 (Ladder pair $(L_{m,u}, L_{m,v})$). *Let $m = m(u, v, v_0)$. As $m \in I(v_0, u)$ and $m \in I(v_0, v)$, pairs (m, u) and (m, v) admit a ladder set. The ladder pair of u and v is $(L_{m,u}, L_{m,v})$.*

If $u \in I(v_0, v)$, then $m = u$. In this case, the ladder pair is the empty set with the ladder set of u and v . Figure 7 represents a triplet u, m, v such that $m = m(u, v, v_0)$. The edges are oriented according to the v_0 -orientation. In this example, the ladder pair of u, v is $(L_{m,u}, L_{m,v})$, with $L_{m,u} = \{E_i, E_j\}$ and $L_{m,v} = \{E_r, E_\ell\}$.

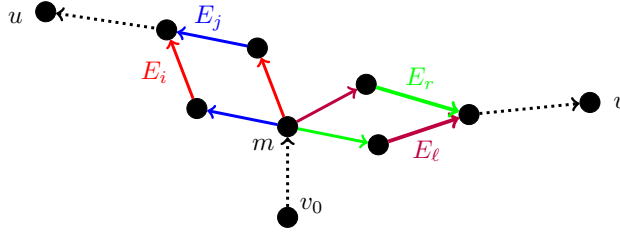


Figure 7: Ladder pairs $L_{m,u} = \{E_i, E_j\}$ and $L_{m,v} = \{E_r, E_\ell\}$.

A shortest (u, v) -path passing through $m = m(u, v, v_0)$ is referred as an “down-up” shortest path in [30]. Indeed, its (u, m) -section is v_0 -oriented “downwards” - edges are oriented towards u - while its (m, v) -section is oriented “upwards” - edges are oriented towards v . The ladder pair of (u, v) indicates the ladder set of both the downwards and the upwards section. They contain the Θ -classes adjacent to m which belong to $\sigma_{m,u}$ and $\sigma_{m,v}$ respectively.

Computation of the labels. We design an algorithm which computes the labels on our median graph G . It determines for each pair (u, L) , $u \in V$ and L is a POF outgoing from u , the length of the longest shortest path starting from u , with an arrival $v \in V$ verifying $u \in I(v_0, v)$, and such that its ladder set is L . We denote by $\varphi(u, L)$ this variable. Put formally, $\varphi(u, L)$ is the maximum distance $d(u, v)$, where $u \in I(v_0, v)$ and $L_{u,v} = L$.

The following theorem identifies a relationship between labels of different vertices.

Theorem 5. *Let $u \in V$, L be a POF outgoing from u and Q be the hypercube with basis u and signature L . We denote by u^+ the opposite vertex of u in Q : u is the basis of Q and u^+ its anti-basis. A vertex $v \neq u^+$ verifies $u \in I(v_0, v)$ and $L_{u,v} = L$ if and only if (i) $u^+ \in I(v_0, v)$ and (ii) for each $E_j \in L_{u^+,v}$, $L \cup \{E_j\}$ is not a POF.*

Proof. Let P_{uu^+} be a shortest (u, u^+) -path made up of edges of the hypercube Q . Let P_{u^+v} be a shortest (u^+, v) -path. All edges of path P_{uu^+} are oriented towards u^+ . Similarly, path P_{u^+v} is

Algorithm 1: The computation of labels $\varphi(u, L)$

```

1: Input: graph  $G$ ,  $\Theta$ -classes  $\mathcal{E}$ , list  $\mathcal{Q}$  of hypercubes (triplets anti-basis, basis, and POF from Lemma 11)
2: Output: Labels  $\varphi(u, L)$  for each vertex  $u \in V$  and POF  $L$  outgoing from  $u$ 
3: Initialize  $\varphi(u, L) \leftarrow 0$  for each  $u \in V$  and  $L$  POF outgoing from  $u$ ;
4:  $\mathcal{Q}^* \leftarrow \text{reverse}(\mathcal{Q})$ ;
5: for  $(u^+, u, L)$  in list  $\mathcal{Q}^*$  do
6:   if  $\varphi(u, L) = 0$  then
7:      $\varphi(u, L) \leftarrow |L|$ ;
8:   endif
9:   for all  $X \subseteq \mathcal{E}^-(u)$ ,  $X \neq \emptyset$  do
10:     $u^- \leftarrow$  basis of the hypercube with anti-basis  $u$  and  $\Theta$ -classes  $X$ ;
11:     $\text{check}^\perp \leftarrow \text{false}$ ;
12:    for all  $E_j \in L$  do
13:      if  $X \cup \{E_j\}$  is a POF then  $\text{check}^\perp \leftarrow \text{true}$ ;
14:    endfor
15:    if not  $\text{check}^\perp$  then  $\varphi(u^-, X) \leftarrow \max\{\varphi(u^-, X), |X| + \varphi(u, L)\}$ ;
16:  endfor

```

oriented towards v as $u^+ \in I(v_0, v)$. Consequently, the concatenation $P_{uu^+} \cdot P_{u^+v}$ is an (u, v) -path whose edges are all oriented towards v . According to Theorem 3, it is a shortest path, otherwise it would pass twice through the same class and an edge of the path would be oriented towards u . So, $u^+ \in I(u, v)$.

We can use the same argument with the triplet (v_0, u, v) , considering a shortest (v_0, u) -path and the shortest (u, v) -path $P_{uu^+} \cdot P_{u^+v}$ which is oriented towards v . We obtain that $u \in I(v_0, v)$.

Finally, we prove that the ladder set of the pair (u, v) is exactly L . Suppose that a class E_j of $L_{u^+,v}$ contains an edge adjacent to u . As $u, u^+ \in \partial H'_j$, the convexity of boundaries (Lemma 4) implies that all vertices of Q are in $\partial H'_j$ because $Q \subseteq I(u, u^+)$. The class E_j defines an isomorphism, so E_j is orthogonal to all classes represented in Q , which corresponds to L . In other words, $L \cup \{E_j\}$ is a POF, a contradiction.

The converse is also true. Let us consider a vertex v satisfying $u \in I(v_0, v)$ and $L_{uv} = L$. We have $u^+ \in I(u, v) \subseteq I(v_0, v)$, therefore the pair (u^+, v) admits a ladder set. Suppose by way a contradiction that there is $E_j \in L_{u^+,v}$ such that $L \cup \{E_j\}$ is a POF. We know that $u^+ \in \partial H'_j$. Lemma 7 implies that the neighbors of u^+ in Q are also in $\partial H'_j$. We can spread this reasoning: the vertices at distance 2 from u^+ in Q are in $\partial H'_j$, etc. Finally, we have $u \in \partial H'_j$. This is a contradiction as the ladder set L_{uv} is L , not $L \cup \{E_j\}$. \square

Theorem 5 is illustrated in Figure 8. It shows a median graph with a basepoint v_0 . Some Θ -classes are colored. Consider the POF $L = \{E_2, E_3\}$ outgoing from u and the ladder set $L_{u^+,x} = \{E_1\}$. As $L \cup \{E_1\}$ is a POF, the ladder set $L_{u,x}$ of pair u, x is not L , but $L_{u,x} = \{E_1, E_2, E_3\}$. However, if we consider the ladder set $L_{u^+,v} = \{E_4, E_5\}$, then we observe that neither $L \cup \{E_4\}$ nor $L \cup \{E_5\}$ is a POF. So, $L_{u,v} = L$ according to Theorem 5. Indeed, one can check that E_2 and E_3 are the only Θ -classes of $\sigma_{u,v}$ which are adjacent to u .

We initialize all labels $\varphi(u, L)$ with 0. We consider the list of hypercubes obtained from Lemma 11. We go through this list in the descending order, *i.e.* starting with hypercubes having the farthest-to- v_0 anti-basis. For each triplet treated, we pick up two of the three elements: the basis u and the sets of Θ -classes of the hypercube that we denote by L . Set L is indeed a POF outgoing from u . Suppose that we are treating the pair (u, L) corresponding to the hypercube Q : all pairs (u^+, L^+) , where u^+ is the anti-basis of Q , have already been considered. Indeed, pairs (u^+, L^+) admit an anti-basis which is farthest to v_0 compared to the anti-basis u^+ of (u, L) .

We give the details of our procedure. Algorithm 1 presents it in pseudocode. First, we verify

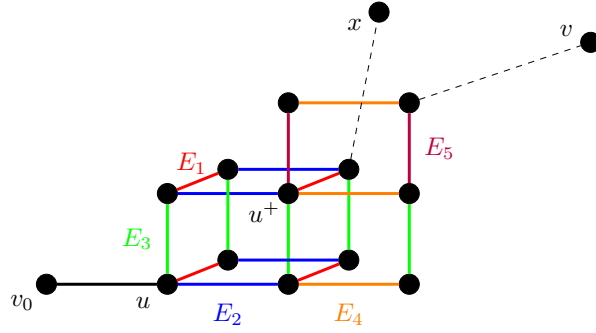


Figure 8: Two vertices u and u^+ respectively basis and anti-basis of a hypercube with classes $L = \{E_2, E_3\}$. POF L is the ladder set of (u, v) , not of (u, x) . Indeed, $L_{u,x} = \{E_1, E_2, E_3\}$.

the value we have for variable $\varphi(u, L)$. If $\varphi(u, L) = 0$, then we put $\varphi(u, L) \leftarrow |L|$ (line 7 in Algorithm 1). This case occurs when the hypercube Q with basis u and POF L is “peripheral”, *i.e.* when the anti-basis u^+ of Q is the vertex v which is the farthest to u and such that $L_{uv} = L$. The distance $d(u, u^+)$ is equal to the dimension of hypercube Q , which is $|L|$. If $\varphi(u, L) > 0$, then we consider that the label provides us with the distance from u to the farthest-to- u vertex satisfying $L_{uv} = L$. Indeed, the anti-basis u^+ of Q has already been treated by the algorithm. We know from Theorem 5 that the value $\varphi(u, L)$ can be deduced from a pair (u^+, L^+) , where L^+ is a POF outgoing from u^+ which does not contain a class forming a POF with L .

Second, we list all subsets of $\mathcal{E}^-(u)$. The idea is to communicate the value $\varphi(u, L)$ to certain vertices which are the bases of hypercubes with anti-basis u . Let us consider one subset, $X \subseteq \mathcal{E}^-(u)$. We know from Lemma 9 that there is an hypercube Q^- with anti-basis u formed by the classes X . We denote by u^- the basis of the hypercube Q^- (line 9). We verify whether there is a class $E_j \in L$ such that $X \cup \{E_j\}$ is a POF (line 12). If the answer is no, then there is a vertex v such that $L_{u^-v} = X$ which is at distance $|X| + \varphi(u, L)$ from u , according to Theorem 5. We update variable $\varphi(u^-, X)$ (line 14):

$$\varphi(u^-, X) \leftarrow \max \{ \varphi(u^-, X), |X| + \varphi(u, L) \}.$$

Theorem 6. *The execution of Algorithm 1 produces labels $\varphi(u, L)$ which are the maximum distances $d(u, v)$, where v satisfying $L_{u,v} = L$.*

Proof. The algorithm treats all pairs (u, L) such that u is a vertex and L is a POF outgoing from u . The first action of the algorithm is to check the current value of $\varphi(u, L)$.

If $\varphi(u, L) = 0$, it means that there is no edge outgoing from u^+ , the anti-basis of the hypercube with basis u and signature L . In this case, u^+ is the farthest-to- u vertex with ladder set L . So, $\varphi(u, L) = d(u, u^+) = |L|$.

If $\varphi(u, L) > 0$, there is an hypercube with basis u^+ and signature L^+ , where $L \cup \{E_i\}$ is not a POF for any $E_i \in L^+$. According to Theorem 5, the algorithm ensures us that the label of (u, L) is correctly computed as we consider all pairs (u^+, L^+) susceptible to influence value $\varphi(u, L)$. \square

We focus now on the running time of Algorithm 1. For each pair (u, L) (said differently for any hypercube of G), we list all subsets of set $\mathcal{E}^-(u)$ which is of size at most d (Lemma 9). For each of these subsets X , we verify whether $X \cup \{E_j\}$ is a POF, where $E_j \in L$. The number of hypercubes is upper-bounded by $2^d n$ (Lemma 10) and the number of subsets X is at most 2^d . Determining whether $X \cup \{E_j\}$ is a POF can be done in $O(1)$ by checking whether vertex u^- is adjacent to an edge of E_j . This argument comes from the convexity of boundaries, already used twice in the proof of Theorem 5. So, the total running time is $O(d2^{2d}n)$.

4.2 Maximum-weighted disjoint sets

In this section, we use the labels computed in Section 4.1 to determine the diameter $\text{diam}(G)$ of the median graph G . The idea is to explore ladder pairs $(L_{m,u}, L_{m,v})$ for all vertices $m \in V$ and to find

the one that maximizes the distance from a vertex u to a vertex v which satisfy $m = m(u, v, v_0)$.

A relationship between the diameter and ladder pairs. We begin with a theorem which allows us to characterize the diametral pair regarding the ladder pairs.

Theorem 7. *Let $m \in V$ and L, L^* be two POFs outgoing from m . Let u, v be two vertices such that m belong to both $I(v_0, u)$ and $I(v_0, v)$. Suppose they fulfil $L_{m,u} = L$ and $L_{m,v} = L^*$. Then, $m \in I(u, v)$ if and only if $L \cap L^* = \emptyset$.*

Proof. First, suppose that $L \cap L^* \neq \emptyset$. Then $\sigma_{m,u} \cap \sigma_{m,v} \neq \emptyset$. As a consequence, $m \notin I(u, v)$: the concatenation of a shortest (u, m) -path with a shortest (m, v) -path does not produce a shortest (u, v) -path because it contains two edges from the same class (Theorem 2).

Second, suppose that $L \cap L^* = \emptyset$. We prove that $m \in I(u, v)$. Let $P_{m,u}$ (resp. $P_{m,v}$) be an arbitrary shortest (m, u) -path (resp. (m, v) -path). If $\sigma_{m,u} \cap \sigma_{m,v} = \emptyset$, then the concatenation of $P_{m,u}$ and $P_{m,v}$ produces a shortest (u, v) -path traversing m , so $m \in I(u, v)$. This is why we suppose, by way of contradiction, that there is a class E_i appearing both in $P_{m,u}$ and $P_{m,v}$. There is one edge (u_i, u_i^*) of E_i in $P_{m,u}$ and one edge (v_i, v_i^*) of E_i in $P_{m,v}$. This notation will be used for any class belonging to both paths, *i.e.* in $\sigma_{m,u} \cap \sigma_{m,v}$.

Instead of considering an arbitrary class E_i admitting one edge in both paths, we denote by E_j a class of $\sigma_{m,u} \cap \sigma_{m,v}$ satisfying a certain property: there is no other class $E_h \in \sigma_{m,u} \cap \sigma_{m,v}$ with both (u_h, u_h^*) closer to m than (u_j, u_j^*) in $P_{m,u}$ and (v_h, v_h^*) closer to m than (v_j, v_j^*) in $P_{m,v}$. Said differently, we fix E_j such that another class of $\sigma_{m,u} \cap \sigma_{m,v}$ that would be closer to m in both paths does not exist. Obviously, such a class E_j exists as set $\sigma_{m,u} \cap \sigma_{m,v}$ is finite.

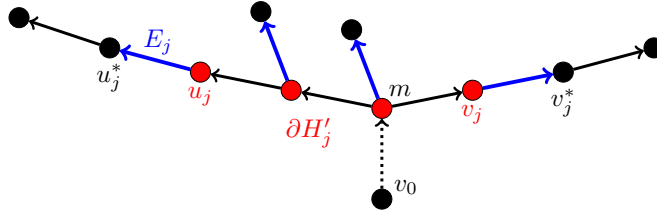


Figure 9: Illustration of the convexity of $\partial H'_j$

Let u_j (resp. v_j) be the vertex in $\partial H'_j$ of path $P_{m,u}$ (resp. $P_{m,v}$). In other words, u_j is the vertex adjacent to the edge of E_j in $P_{m,u}$ which is the closest to v_0 , so the closest to m . With the definition of class E_j , we have $m \in I(u_j, v_j)$. Indeed, the concatenation of the (m, u_j) -section of $P_{m,u}$ and of the (m, v_j) -section of $P_{m,v}$ produces a shortest (u_j, v_j) -path passing through m as it does not contain two edges of the same class. As $u_j, v_j \in \partial H'_j$ and $\partial H'_j$ is convex, then vertex m is adjacent to an edge of E_j : $m \in \partial H'_j$. Figure 9 illustrates the latter assertion: the edges of E_j are drawn in blue, the vertices of $\partial H'_j$ in red. As $E_j \in \sigma_{m,u} \cap \sigma_{m,v}$, we have $E_j \in L_{m,u} \cap L_{m,v}$ which is a contradiction. \square

Thanks to Theorem 7, we establish a relationship between the diameter $\text{diam}(G)$ and the ladder pairs. For $m \in V$, let $\Upsilon(m)$ be the distance of the longest shortest (u, v) -path such that $m = m(u, v, v_0)$. The maximum $\Upsilon(G)$ over all $\Upsilon(m)$, $m \in V$ is naturally: $\Upsilon(G) = \text{diam}(G)$.

Corollary 1. *For any $m \in V$, value $\Upsilon(m)$ is the maximum of $\varphi(m, L) + \varphi(m, L^*)$ for two POFs L and L^* outgoing from m and with an empty intersection. Formally,*

$$\Upsilon(m) = \max_{\substack{L \cap L^* = \emptyset \\ \text{outgoing from } m}} \varphi(m, L) + \varphi(m, L^*). \quad (2)$$

The diameter of G can thus be written:

$$\text{diam}(G) = \Upsilon(G) = \max_{m \in V} \max_{\substack{L \cap L^* = \emptyset \\ \text{outgoing from } m}} \varphi(m, L) + \varphi(m, L^*). \quad (3)$$

Proof. Let $m \in V$ and u, v be the pair of vertices satisfying $m = m(u, v, v_0)$ which maximizes $d(u, v)$. We know from Theorem 7 that the ladder sets $L_{m,u}$ and $L_{m,v}$ are outgoing from m and have an empty intersection. We have $\varphi(m, L_{m,u}) = d(m, u)$, otherwise there would exist another vertex u^* , where $d(m, u^*) > d(m, u)$ and $L_{m,u} = L_{m,u^*}$, and we would obtain $d(u^*, v) = d(u^*, m) + d(m, v) > d(u, v)$, a contradiction. For the same reason, $\varphi(m, L_{m,v}) = d(m, v)$. So, we know that there is a pair $L_{m,u} \cap L_{m,v} = \emptyset$ of POFs outgoing from m such that $d(u, v) = \varphi(m, L_{m,u}) + \varphi(m, L_{m,v})$.

Now, suppose that there is another pair $L \cap L^* = \emptyset$ outgoing from m such that $d(u, v) < \varphi(m, L) + \varphi(m, L^*)$. Theorem 7 implies that there are two vertices x, y verifying $d(m, x) = \varphi(m, L)$, $d(m, y) = \varphi(m, L^*)$, and $m = m(x, y, v_0)$. So, $d(x, y) > d(u, v)$. As $m = m(x, y, v_0)$, we have a contradiction with the definition of u, v . In brief, $L_{m,u}, L_{m,v}$ is the pair of disjoint POFs outgoing from m which maximizes the sum of labels. Equation (2) holds.

Equation (3) is a direct consequence of Equation (2) because the diameter is the maximum of $\Upsilon(m)$ over all vertices $m \in V$ by definition. \square

According to this corollary, a way to obtain the diameter is to find the triplet (m, L, L^*) which maximizes $\varphi(m, L) + \varphi(m, L^*)$, where L and L^* are disjoint POFs outgoing from m . Our idea is to determine $\Upsilon(m)$ for each $m \in V$ and then to identify the maximum of these values. We thus propose an algorithm returning all values $\Upsilon(m)$ in linear time for constant d .

Computation of $\Upsilon(m)$. We fix some $m \in V$. Let \mathcal{L}_m be the set of POFs outgoing from m . Its cardinality is denoted by N_m . We know that \mathcal{L}_m is closed under subsets and, for any $L \in \mathcal{L}_m$, $|L| \leq d$. A positive integer $\varphi(m, L)$ is associated with any of these sets. From now on, we denote them by $\omega(L)$ to be concise: $\omega(L) = \varphi(m, L)$. The goal is to find the pair $L, L^* \in \mathcal{L}_m$, $L \cap L^* = \emptyset$ maximizing $\omega(L) + \omega(L^*)$. We call this problem MAXIMUM-WEIGHTED DISJOINT SETS (MWDS).

Definition 6 (Maximum-weighted disjoint sets).

Input: Elements \mathcal{E} , collection \mathcal{L}_m of sets of elements, weights $\omega : \mathcal{L}_m \rightarrow \mathbb{N}^+$, parameter $d \in \mathbb{N}$. For each $L \in \mathcal{L}_m$, $|L| \leq d$.

Output: Pair $L, L^* \in \mathcal{L}_m$, $L \cap L^* = \emptyset$ which maximizes $\omega(L) + \omega(L^*)$.

We design an algorithm solving MWDS in time $O(2^{d \log d} N_m)$. Before describing it, we need some notation. Given $L \in \mathcal{L}_m$, we denote by $\text{op}(L)$ the *opposite* of set L , i.e. the set in \mathcal{L}_m verifying $L \cap \text{op}(L) = \emptyset$ with maximum weight $\omega(\text{op}(L))$. Our algorithm will consist in identifying the opposite $\text{op}(L)$ of each $L \in \mathcal{L}_m$.

We construct a tree T_m which allows us to compute all opposites. To avoid confusions, the vertices of T_m are called *nodes*. Nodes of T_m are indexed with POFs in \mathcal{L}_m and edges of T_m are indexed with Θ -classes. For any node $a \in T_m$, we denote by $R(a)$ the union of indices of the edges which are on the simple path from the root of T_m to node a . With our construction, we announce that any set $R(a)$ will be a POF.

We begin with an iterative presentation of our algorithm. We identify first the set $L_0 \in \mathcal{L}_m$ with the maximum weight. The running time of such step is $O(N_m)$. The root of our tree is indexed with L_0 (line 17 of Algorithm 2). At this step, we know that all sets $L \in \mathcal{L}_m$ verifying $L_0 \cap L = \emptyset$ admit L_0 as their opposite: $\text{op}(L) = L_0$. The remaining sets are the ones having a nonempty intersection with L_0 .

For each class $E_{i_0} \in L_0$, we add a child to L_0 and the edge of T_m that connects the root with this child is indexed with E_{i_0} . For example, suppose that $L_0 = \{E_i, E_j\}$. Then, the root has two children. The two edges connecting L_0 to its children are respectively indexed with E_i and E_j (Figure 10a). The idea is that if the intersection of a given set L with L_0 is $\{E_i\}$ for example, then traverse the branch indexed with E_i to find the opposite of L . If the intersection is $\{E_i, E_j\}$, then traverse arbitrarily one of the two branches E_i and E_j .

The next step consists in indexing the children of L_0 . We consider one of them being connected to L_0 by a branch with index $E_{i_0} \in L_0$. We denote this node by a_1 . We compute the set $L_1 \in \mathcal{L}_m$ of maximum weight which does not contain any class of $R(a_1) = \{E_{i_0}\}$. We fix L_1 as the index of a_1 . If we go back to our example, with $E_{i_0} = E_i$, then L_1 is the maximum-weighted POF in \mathcal{L}_m that does not contain E_i . Suppose that $L_1 = \{E_j, E_h, E_r\}$. At this moment, we have treated the sets $L \in \mathcal{L}_m$ such that $L_0 \cap L = \emptyset$: their opposite is L_0 . With this child a_1 , we aim at finding the opposite of sets verifying $E_i \in L$. For example, for $L = \{E_i, E_\ell\}$, its opposite is indeed $\text{op}(L) = L_1 = \{E_j, E_h, E_r\}$ because $L \cap L_1 = \emptyset$ and L_1 is the maximum-weighted set which does

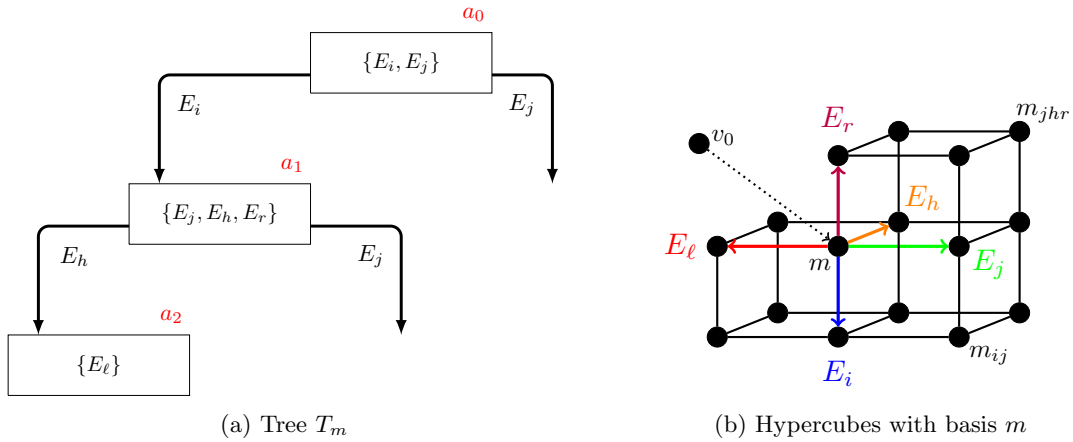


Figure 10: An example of tree T_m which allows us to compute all opposites in \mathcal{L}_m .

not contain E_i . Unfortunately, if $L = \{E_i, E_h\}$, then L and L_1 are not disjoint so we still have to find the opposite of $\{E_i, E_h\}$. Put formally, we know that L_0 is the opposite of any $L \in \mathcal{L}_m$ satisfying $L \cap L_0 = \emptyset$. Moreover, we know that a child L_1 of L_0 , connected via a branch $E_{i_0} \in L_0$, is the opposite of $L \in \mathcal{L}_m$ satisfying $E_{i_0} \in L \cap L_0$ and $L \cap L_1 = \emptyset$.

We pursue the construction of the tree T_m . The node a_1 indexed by L_1 admits one child for each class $E_{i_1} \in L_1$ whose union with $R(a_1)$ form a POF. In our example, $R(a_1) = \{E_i\}$ and $L_1 = \{E_j, E_h, E_r\}$, so if for example $\{E_i, E_h\}$ is POF, then we add a child (say a_2) to a_1 and the index of the edge (a_1, a_2) is E_h . The index of node a_2 is the POF of maximum weight which has an empty intersection with $R(a_2)$. In our example, $R(a_2) = \{E_i, E_h\}$, see Figure 10a.

Algorithm 2 presents a recursive view of the construction of the tree. Symbol $\#$ refers to the comments, outside the pseudocode. Figure 10 shows an example of tree T_m and besides, an example of median graph where only the hypercubes with basis m are represented. For each class, its edge adjacent to m is colored and oriented in accordance with the v_0 -orientation. As the root of T_m is indexed with $\{E_i, E_j\}$, then we can affirm that the vertex v satisfying $m \in I(v_0, v)$ which maximizes $d(m, v)$ verifies $L_{mv} = \{E_i, E_j\}$. Therefore, the longest shortest (m, v) -path, such that $m \in I(v_0, v)$, passes through vertex m_{ij} . As an example, the opposite of $\{E_i\}$ is $\{E_j, E_h, E_r\}$. So, if a diametral pair (u, v) is such that $m = m(u, v, v_0)$ and $L_{mu} = \{E_i\}$, then the diameter is given by value $\varphi(m, \{E_i\}) + \varphi(m, \{E_j, E_h, E_r\})$ and a diametral path passes through vertex m_{jhr} .

Once the computation of the tree T_m is completed, the index L_a of node $a \in T_m$ is the opposite of any $L \in \mathcal{L}_m$ that contains all classes of $R(a)$ but no class of L_a . Moreover, there is one child of node $a \in T_m$ for each class of its index L_a whose union with $R(a)$ is a POF (line 7). Indeed, if some $L \in \mathcal{L}_m$ contains $R(a)$ but also certain classes of the index L_a of a , then we have to go down the tree through a branch indexed by an arbitrary class E_i in $L_a \cap L$ to find the opposite of L . The following lemma ensures us that all opposites in \mathcal{L}_m can be identified in the tree T_m .

Lemma 12. *Let a be a node of T_m , L_a its index, and L some POF in \mathcal{L}_m . Assume that $R(a) \subseteq L$. If $L_a \cap L = \emptyset$, then $\text{op}(L) = L_a$. Otherwise, there exists a child, denoted by $a[E_i]$, where the index E_i of edge $(a, a[E_i])$ belongs to $L_a \cap L$.*

Proof. By definition, the POF L_a is the maximum-weighted POF which has no intersection with $R(a)$. If L_a has no intersection with L , then we cannot find another POF disjoint from $L \supseteq R(a)$ with a greater weight than $\omega(L_a)$, so $\text{op}(L) = L_a$ in this case. Otherwise, if there is some $E_i \in L_a \cap L$, we can say that $\{E_i\} \cup R(a) \subseteq L$ is a POF as its superset L . Consequently, as $E_i \in L_a$, there is an edge between a and one of its children which has index E_i . \square

We determine the opposite $\text{op}(L)$ of each $L \in \mathcal{L}_m$ with a tree search in T_m . If $L \cap L_0 = \emptyset$, then $\text{op}(L) = L_0$, else traverse the branch with an index in $L \cap L_0$, etc. Each node a visited in this descent verifies $R(a) \subseteq L$. If $L_a \cap L = \emptyset$, then $\text{op}(L) = L_a$ (Lemma 12). Otherwise, at least one child of a can be visited to pursue the descent, via a branch $E_i \in L$ which exists according to

Algorithm 2: Construction of the tree T_m providing us with all opposites.

```

1: Input: Vertex  $m$ ,  $\Theta$ -classes  $\mathcal{E}$ , set  $\mathcal{L}_m$  of POFs and  $\omega : \mathcal{L}_m \rightarrow \mathbb{N}^+$ 
2: Output: Tree  $T_m$ 
3: def children( $T, a$ ) :
4:    $L_a \leftarrow$  index of node  $a$ ;
5:   for all  $E_i \in L_a$  do
6:     if  $\{E_i\} \cup R(a)$  is a POF then
7:       add a child  $a[E_i]$  of  $a$  in  $T$ ; Index  $(a, a[E_i])$  with  $E_i$ ;
8:        $L_{a[E_i]} \leftarrow \underset{\substack{L \in \mathcal{L}_m \\ L \cap R(a[E_i]) = \emptyset}}{\operatorname{argmax}} \omega(L)$ ; # here,  $R(a[E_i]) = \{E_i\} \cup R(a)$ 
9:       add index  $L_{a[E_i]}$  to node  $a[E_i]$ ;
10:    endif
11:  endfor
12:  for all children  $a[E_i]$  of  $a$  in  $T$  do
13:    | children( $T, a[E_i]$ );
14:  endfor
15: enddef
16:  $L_0 \leftarrow \operatorname{argmax}_{L \in \mathcal{L}_m} \omega(L)$ ;
17:  $a_0 \leftarrow$  node with index  $L_0$ ;  $T_m \leftarrow$  single node  $a_0$ ; # here,  $R(a) = \emptyset$ 
18: children( $T_m, a_0$ );

```

Lemma 12. The depth of T_m is at most d because $R(a)$ is a POF for any $a \in T_m$ and the cardinality of any POF is at most d . The tree search for each L costs only $O(d^2)$ because we visit at most d nodes (depth) and checking the intersections with all indices L_a costs $O(d)$. The operation consisting in finding all opposites in \mathcal{L}_m takes thus $O^*(N_m)$, where O^* neglects polynomials of d . The computation of $\Upsilon(m)$ is complete: we pick up the pair $(L, \operatorname{op}(L))$ with maximum weight.

Running time and discussion. We provide an upper bound of the number of vertices of T_m . First, the depth of T_m is at most d . Second, each node has at most d children. Therefore, the number of nodes is at most $d! = O(2^{d \log d})$. Computing the index of each of its nodes (line 8 of Algorithm 2) takes $O(N_m)$. The construction of tree T_m takes $O^*(2^{d \log d} N_m)$.

The running time for determining $\Upsilon(m)$ - construction of the tree and computation of the opposites - is $O^*(2^{d \log d} N_m)$. Eventually, we pick up the vertex m maximizing $\Upsilon(m)$ and the value obtained is $\operatorname{diam}(G)$ according to Corollary 1. The running time of the algorithm is thus $O^*(2^{d(\log d + 1)} n)$ because $\sum_{m \in V} N_m = \alpha(G) \leq 2^d n$.

Our algorithm outputs the diameter. However, we may aim at returning a diametral pair. To do so, we can simply add an extra label $\mu(u, L)$ in order to obtain a labeling pair $(\mu(u, L), \varphi(u, L))$, where $\mu(u, L)$ is a vertex which is at distance $\varphi(u, L)$ from $u \in I(v_0, \mu(u, L))$ with ladder set L . Two slight modifications must be done in Algorithm 1 presented in Sections 4.1. First, when $\varphi(u, L) = 0$ (line 7), we initialize the vertex $\mu(u, L)$ with the anti-basis u^+ . Second, the update of $\varphi(u^-, X)$ in line 14 is replaced by an update of both the vertex $\mu(u^-, X)$ and the value $\varphi(u^-, X)$. If $|X| + \varphi(u, L) > \varphi(u^-, X)$, then $\mu(u^-, X) \leftarrow \mu(u, L)$.

5 Computing all eccentricities in linear time for dimension $d = O(1)$

We use the computations of Section 4 to determine the eccentricity of each vertex of G . We remind that, in time $O(2^{O(d \log d)} n)$, we obtained the following labels for any pair (u, L) , where $u \in V$ and L is a POF outgoing from u .

- $\varphi(u, L)$: distance of the longest shortest path starting from vertex u with ladder set L ,

- $\text{op}_u(L)$: ladder set L^* outgoing from u which has an empty intersection with L and maximizes $\varphi(u, L^*)$.

We proceed in two steps in this section. First, we define the concept of *milestones* which consist in particular vertices of any interval $I(u, v)$ satisfying $u \in I(v_0, v)$. Second, we compute new labels $\psi(u, X)$, where $u \in V$ and X is a POF arriving in u . Their definition depend on the notion of milestone. Finally, we establish a relationship between the eccentricity $\text{ecc}(u)$ of $u \in V$ and both labels $\varphi(u, L)$ and $\psi(u, X)$.

5.1 Milestones

We consider two vertices u, v such that $u \in I(v_0, v)$, said differently $u = m(u, v, v_0)$. The notion of *milestone* is defined recursively.

Definition 7 (Milestones $\Pi(u, v)$). *Let $L_{u,v}$ be the ladder set of u, v and u^+ be the anti-basis of the hypercube with basis u and signature $L_{u,v}$. If $u^+ = v$, then pair u, v admits two milestones: $\Pi(u, v) = \{u, v\}$. Otherwise, the set $\Pi(u, v)$ is the union of $\Pi(u^+, v)$ with vertex u : $\Pi(u, v) = \{u\} \cup \Pi(u^+, v)$.*

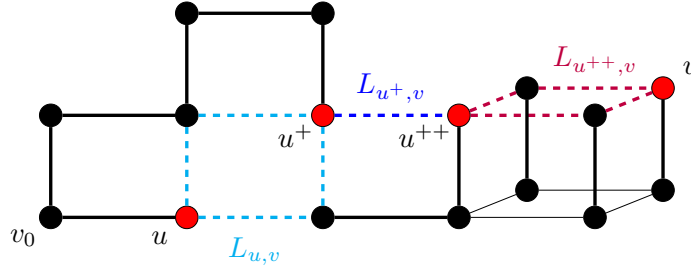


Figure 11: A pair u, v with $u \in I(v_0, v)$ and its milestones $\Pi(u, v)$ in red.

The milestones are the successive anti-bases of the hypercubes formed by the vertices and ladder sets traversed from u to v . Moreover, it contains both u and v . Concretely, the first milestone is u , the second is the anti-basis u^+ of the hypercube characterized by $(u, L_{u,v})$. The third one is the anti-basis u^{++} of the hypercube with basis u^+ and Θ -classes $L_{u^+,v}$, etc. All milestones are in the interval of u and v : $\Pi(u, v) \subseteq I(u, v)$.

Figure 11 represents the same graph than in Figure 6 and shows the milestones $\Pi(u, v) = \{u, u^+, u^{++}, v\}$. The hypercubes with the following pair basis-signature are highlighted with dashed edges: $(u, L_{u,v})$, $(u^+, L_{u^+,v})$, and $(u^{++}, L_{u^{++},v})$.

We say that the milestone in $\Pi(u, v)$ which is different from v but the closest to it is called the *penultimate milestone*. We denote it by $\bar{\pi}(u, v)$. For example, u^{++} is the penultimate milestone of $\Pi(u, v)$ in Figure 11. Furthermore, we denote by $\bar{L}_{u,v}$ the Θ -classes of the hypercube with basis $\bar{\pi}(u, v)$ and anti-basis v . In Figure 11, $\bar{L}_{u,v} = L_{u^{++},v}$.

A consequence of Theorem 5 is that, for two consecutive milestones in $\Pi(u, v)$, say u and u^+ w.l.o.g, then $L_{u,v}$ and $L_{u^+,v}$ verify the following property: for any class $E_i \in L_{u^+,v}$, then $L_{u,v} \cup \{E_i\}$ is not a POF. We establish a property dealing with the penultimate milestone.

Theorem 8. *Let $u, v \in V$ and $u \in I(v_0, v)$. Let L be a POF outgoing from v and w the anti-basis of hypercube (v, L) . The following propositions are equivalent:*

- vertex v is the penultimate milestone of (u, w) : $\bar{\pi}(u, w) = v$,*
- the milestones of (u, w) are the milestones of (u, v) with w : $\Pi(u, w) = \Pi(u, v) \cup \{w\}$,*
- for any class $E_i \in L$, then $\bar{L}_{u,v} \cup \{E_i\}$ is not a POF.*

Proof. Claim: $\sigma_{u,w} = \sigma_{u,v} \cup L$. As $u \in I(v_0, v)$ and $v \in I(v_0, w)$, we have $u \in I(v_0, w)$ and $v \in I(u, w)$. The concatenation of a shortest (u, v) -path with a shortest (v, w) -path produces a shortest (u, w) -path. According to Theorem 2, $\sigma_{u,w} = \sigma_{u,v} \cup \sigma_{v,w}$. As w is the anti-basis of hypercube (v, L) , then $\sigma_{v,w} = L$, which proves the claim.

(i) \Leftrightarrow (ii). Suppose that v is the penultimate milestone in $\Pi(u, w)$. From the previous claim, we know that $\sigma_{u,w} = \sigma_{u,v} \cup L$. Consider a milestone in $\Pi(u, w)$ arriving before v , i.e. $p \in \Pi(u, w) \setminus \{v, w\}$, and suppose that some class $E_i \in L$ belongs to the ladder set $L_{p,w}$. The milestone after p in $\Pi(u, w)$ is thus in $\partial H_i''$. By convexity of $\partial H_i''$, all milestones arriving after p are in $\partial H_i''$. This is a contradiction as an edge of E_i is outgoing from v , so $v \in \partial H_i'$. So, no class of L appears in the ladder sets $L_{p,w}$ before milestone v . The milestones of pair (u, w) are built with the classes in $\sigma_{u,v}$ until we arrive at v : $\Pi(u, w) = \Pi(u, v) \cup \{w\}$. Conversely, (ii) \Rightarrow (i) is trivial.

(ii) \Leftrightarrow (iii). The direct side is a consequence of Theorem 5. As $\bar{\pi}(u, v) \in I(v_0, v)$ and $v \in I(v_0, w)$, we have $\bar{\pi}(u, v) \in I(v_0, w)$. Vertex $\bar{\pi}(u, v)$ belongs to $\Pi(u, w)$, so the ladder set of $\bar{\pi}(u, v)$ and w is $\bar{L}_{u,v}$. We apply Theorem 5 to vertices $\bar{\pi}(u, v)$ and w : for any $E_i \in L_{v,w} = L$, set $\bar{L}_{u,v} \cup \{E_i\}$ is not a POF.

Suppose now that for any $E_i \in L$, set $\bar{L}_{u,v} \cup \{E_i\}$ is not a POF. The Θ -classes in L are outgoing from v . Assume a class E_i of L is outgoing from some milestone p in $\Pi(u, v)$ different from v : $p \in \Pi(u, v) \setminus \{v\}$. By convexity of $\partial H_i'$, as $\{p, v\} \subseteq \partial H_i'$, we have $\bar{\pi}(u, v) \in \partial H_i'$. More generally, all vertices of the hypercube with basis $\bar{\pi}(u, v)$ and signature $\bar{L}_{u,v}$ are in $\partial H_i'$ for the same reason. Class E_i form an isomorphism between $\partial H_i'$ and $\partial H_i''$: there is an hypercube with basis $\bar{\pi}(u, v)$ and classes $\bar{L}_{u,v} \cup \{E_i\}$. This is a contradiction as $\bar{L}_{u,v} \cup \{E_i\}$ is not a POF.

We just showed that no class of L is adjacent to a milestone in $\Pi(u, v) \setminus \{v\}$. As $\sigma_{u,w} = \sigma_{u,v} \cup L$, the milestones of $\Pi(u, w)$ are exactly the milestones of $\Pi(u, v)$ until we arrive at v . Indeed, the ladder set $L_{p,w}$ for any milestone $p \in \Pi(u, v) \setminus \{v\}$ cannot contain a class of L . At vertex v , the ladder set $L_{v,w}$ is L , so the next milestone is w itself. \square

This result is the keystone to compute the labels $\psi(u, X)$ and, finally, all eccentricities.

5.2 Labels $\psi(u, X)$ and eccentricities

Let X be a POF ingoing to some vertex u and u^- be the basis of the hypercube with anti-basis u and classes X . The label $\psi(u, X)$ is the maximum distance $d(u, v)$ we can obtain with a vertex v satisfying the following properties:

- $m = m(u, v, v_0) \neq u$,
- vertex u^- is the penultimate milestone of pair m, u : $u^- = \bar{\pi}(m, u)$.

We present an inductive algorithm determining labels $\psi(u, X)$. We list the hypercubes of G in the order given by list \mathcal{Q} computed in Lemma 11 and evoked in Algorithm 1. Concretely, the hypercubes admitting v_0 as their basis are listed first. The ‘‘peripheral’’ hypercubes are listed last. For each of these hypercubes, we pick up their anti-basis u and their signature X and then we determine $\psi(u, X)$ as explained below.

Let $Q_0 \in \mathcal{Q}$ be an hypercube with basis v_0 . This is the base case of our induction. We denote its anti-basis by v_0^+ and its classes by X_0 . Figure 12 presents an example where $X_0 = \{E_i, E_j\}$ and its opposite $\text{op}_{v_0}(X_0) = \{E_\ell, E_r\}$. We aim at finding the maximum distance $d(v_0^+, v)$ with a vertex v verifying $m = m(v_0^+, v, v_0) \neq v_0^+$ and $v_0 = \bar{\pi}(m, v_0^+)$. Necessarily, $m = v_0$ as $v_0 \in I(v_0^+, v)$. Therefore, we seek the longest shortest path starting from v_0 such that, concatenated with a shortest (v_0, v_0^+) path, it produces a shortest path with v_0^+ as an extremity. As the distance between v_0 and v_0^+ is $|X_0|$, and according to Theorem 7, we have:

$$\psi(v_0^+, X_0) = |X_0| + \varphi(v_0, \text{op}_{v_0}(X_0)) \quad (4)$$

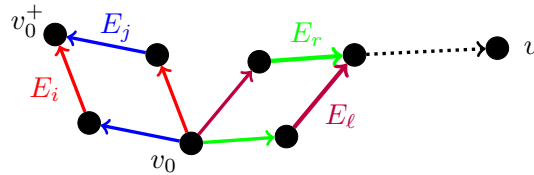


Figure 12: The hypercubes with basis v_0 and signature $X_0 = \{E_i, E_j\}$ and $\text{op}_{v_0}(X_0) = \{E_\ell, E_r\}$

Let $Q \in \mathcal{Q}$ be an hypercube, with basis u^- , anti-basis u and classes X . We suppose that the labels ψ of vertex u^- have already been computed. We are looking for the maximum distance $d(u, v)$ with a vertex v verifying $m = m(u, v, v_0) \neq u$ and $u^- = \bar{\pi}(m, u)$. We distinguish two cases.

First, we assume $m = u^-$. As for the base case (Equation (4)), $\psi(u, X) = |X| + \varphi(u^-, \text{op}_{u^-}(X))$.

Second, we assume that $m \neq u^-$. Set $\Pi(m, u)$ admits at least three milestones: m , u^- , and u . Let X^- be the POF ingoing to u^- which is the ladder set in $\Pi(m, u)$ of the milestone just before u^- . According to Theorem 8, vertex u^- is the penultimate milestone of (m, u) if and only if $X^- \cup \{E_i\}$ is not a POF, for each $E_i \in X$. For this reason, value $\psi(u, X)$ can be expressed as:

$$\psi(u, X) = \max_{\substack{X^- \text{ POF ingoing to } u^- \\ \forall E_i \in X, X^- \cup \{E_i\} \text{ not POF}}} |X| + \psi(u^-, X^-)$$

Our algorithm consists in taking the maximum value between the two cases. In this way, we obtain all labels $\psi(u, X)$ in time $O^*(2^d \alpha(G)) = O^*(2^{2d} n)$ because we consider each hypercube of the graph and list the POFs ingoing to their basis.

Eventually, we show how to compute all eccentricities in function of labels $\varphi(u, L)$ and $\psi(u, X)$.

Theorem 9. *The eccentricity of vertex $u \in V$ is the maximum between all values $\varphi(u, L)$ and also all values $\psi(u, X)$. Formally,*

$$\text{ecc}(u) = \max \left\{ \max_{\substack{L \text{ POF} \\ \text{outgoing from } u}} \varphi(u, L), \max_{\substack{X \text{ POF} \\ \text{ingoing to } u}} \psi(u, X) \right\}$$

Proof. Let v be a vertex in G such that $\text{ecc}(u) = d(u, v)$. If $m = m(u, v, v_0) = u$, then $u \in I(v_0, v)$ and value $d(u, v)$ is given by the label $\varphi(u, L)$, where L is the ladder set $L_{u,v}$. Conversely, each $\varphi(u, L')$ is the distance between u and some vertex v' satisfying $u \in I(v_0, v')$. Therefore, the maximum value of $\varphi(u, L)$ is the maximum distance $d(u, v)$ such that $m = u$.

If $m \neq u$, let u^- be the penultimate milestone in $\Pi(m, u)$ and X be the classes of the hypercube with basis u^- and anti-basis u . In this case, distance $d(u, v)$ is given by label $\psi(u, X)$. So, the maximum value $\psi(u, X)$ is the maximum distance $d(u, v)$ such that $m \neq u$. \square

There are as many labels $\varphi(u, L)$ as hypercubes and the number of labels $\psi(u, X)$ is upper-bounded by $2^d n$. In brief, computing all eccentricities takes $2^{O(d \log d)} n$: the most expensive operation is still Algorithm 2 which determines the opposites of any pair (u, L) .

6 Conclusion

The main contribution of this article is an algorithm determining the diameter, the radius, and all eccentricities in median graphs in linear time when $d = O(1)$. A natural question is whether the techniques we used can be extended to propose an algorithm with a subquadratic running time for all median graphs.

We believe some of the notions we proposed characterize shortest paths in median graphs and are powerful tools to determine the diameter and, more generally, all eccentricities. Ladder sets, ladder pairs but also milestones allow us to focus on certain distances instead of considering all pairs of vertices, as with a multiple BFS. The dependence of distances between two ‘‘adjacent’’ ladder sets (Theorem 5) and the characterization of the diameter via ladder pairs (Corollary 1) are results which are likely to be used again in future research for median graphs.

The design of a linear-time algorithm for the diameter on median graphs seems compromised. Indeed, in [25], the authors put in evidence the difficulty to design a linear-time algorithm computing the diameter for graphs with unbounded distance VC-dimension, which is at least d on median graphs. Nevertheless, in our opinion, the design of a subquadratic algorithm is a challenging line of research.

References

- [1] A. Abboud, F. Grandoni, and V. V. Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *Proc. of SODA*, pages 1681–1697, 2015.
- [2] A. Abboud, V. V. Williams, and J. R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proc. of SODA*, pages 377–391, 2016.
- [3] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.
- [4] S. P. Avann. Metric ternary distributive semi-lattices. *Proc. Amer. Math. Soc.*, 12:407–414, 1961.
- [5] K. Balakrishnan, B. Bresar, M. Changat, S. Klavzar, M. Kovse, and A. R. Subhamathi. Computing median and antimedian sets in median graphs. *Algorithmica*, 57(2):207–216, 2010.
- [6] H. Bandelt. Retracts of hypercubes. *Journal of Graph Theory*, 8(4):501–510, 1984.
- [7] H. Bandelt and J. Barthélemy. Medians in median graphs. *Discret. Appl. Math.*, 8(2):131–142, 1984.
- [8] H. Bandelt and V. Chepoi. Metric graph theory and geometry: a survey. *Contemp. Math.*, 453:49–86, 2008.
- [9] H. Bandelt, V. Chepoi, A. W. M. Dress, and J. H. Koolen. Combinatorics of lopsided sets. *Eur. J. Comb.*, 27(5):669–689, 2006.
- [10] H. Bandelt, V. Chepoi, and D. Eppstein. Combinatorics and geometry of finite and infinite squaregraphs. *SIAM J. Discret. Math.*, 24(4):1399–1440, 2010.
- [11] H. Bandelt, L. Quintana-Murci, A. Salas, and V. Macaulay. The fingerprint of phantom mutations in mitochondrial dna data. *Am. J. Hum. Genet.*, 71:1150–1160, 2002.
- [12] H. J. Bandelt, P. Forster, B. C. Sykes, and M. B. Richards. Mitochondrial portraits of human populations using median networks. *Genetics*, 141(2):743–753, 1995.
- [13] J. Barthélemy and J. Constantin. Median graphs, parallelism and posets. *Discret. Math.*, 111(1-3):49–63, 1993.
- [14] L. Bénéteau, J. Chalopin, V. Chepoi, and Y. Vaxès. Medians in median graphs and their cube complexes in linear time. In *Proc. of ICALP*, volume 168, pages 10:1–10:17, 2020.
- [15] G. Birkhoff and S. A. Kiss. A ternary operation in distributive lattices. *Bull. Amer. Math. Soc.*, 53:745–752, 1947.
- [16] M. Borassi, P. Crescenzi, M. Habib, W. A. Kosters, A. Marino, and F. W. Takes. Fast diameter and radius BFS-based computation in (weakly connected) real-world graphs: With an application to the six degrees of separation games. *Theor. Comput. Sci.*, 586:59–80, 2015.
- [17] B. Bresar, S. Klavzar, and R. Skrekovski. Roots of cube polynomials of median graphs. *Journal of Graph Theory*, 52(1):37–50, 2006.
- [18] S. Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. In *Proc. of SODA*, pages 2143–2152, 2017.
- [19] S. Chechik, D. H. Larkin, L. Roditty, G. Schoenebeck, R. E. Tarjan, and V. V. Williams. Better approximation algorithms for the graph diameter. In *Proc. of SODA*, pages 1041–1052, 2014.
- [20] V. Chepoi. Graphs of some CAT(0) complexes. *Adv. Appl. Math.*, 24(2):125–179, 2000.
- [21] V. Chepoi, F. F. Dragan, and Y. Vaxès. Center and diameter problems in plane triangulations and quadrangulations. In *Proc. of SODA*, pages 346–355, 2002.
- [22] V. Chepoi, A. Labourel, and S. Ratel. Distance labeling schemes for cube-free median graphs. In *Proc. of MFCS*, volume 138, pages 15:1–15:14, 2019.
- [23] D. G. Corneil, F. F. Dragan, and E. Köhler. On the power of BFS to determine a graph’s diameter. *Networks*, 42(4):209–222, 2003.

- [24] G. Ducoffe. Isometric embeddings in trees and their use in the diameter problem. *CoRR*, abs/2010.15803, 2020.
- [25] G. Ducoffe, M. Habib, and L. Viennot. Diameter computation on H -minor free graphs and graphs of bounded (distance) VC-dimension. In *Proc. of SODA*, pages 1905–1922, 2020.
- [26] J. Hagauer, W. Imrich, and S. Klavzar. Recognizing median graphs in subquadratic time. *Theor. Comput. Sci.*, 215(1-2):123–136, 1999.
- [27] R. Hammack, W. Imrich, and S. Klavzar. *Handbook of Product Graphs, Second Edition*. CRC Press, Inc., 2011.
- [28] G. Handler. Minimax location of a facility in an undirected tree graph. *Transp. Sci.*, 7:287–293, 1973.
- [29] W. Imrich, S. Klavzar, and H. M. Mulder. Median graphs and triangle-free graphs. *SIAM J. Discret. Math.*, 12(1):111–118, 1999.
- [30] S. Klavzar and H. M. Mulder. Median graphs: Characterizations, location theory and related structures. *J. Combin. Math. Combin. Comput.*, 30:103–127, 1999.
- [31] S. Klavzar, H. M. Mulder, and R. Skrekovski. An Euler-type formula for median graphs. *Discret. Math.*, 187(1-3):255–258, 1998.
- [32] M. Kovse. Complexity of phylogenetic networks: counting cubes in median graphs and related problems. *Analysis of complex networks: From Biology to Linguistics*, pages 323–350, 2009.
- [33] C. Magnien, M. Latapy, and M. Habib. Fast computation of empirically tight bounds for the diameter of massive graphs. *ACM J. Exp. Algorithmics*, 13, 2008.
- [34] F. R. McMorris, H. M. Mulder, and F. S. Roberts. The median procedure on median graphs. *Discret. Appl. Math.*, 84(1-3):165–181, 1998.
- [35] H. M. Mulder and A. Schrijver. Median graphs and Helly hypergraphs. *Discret. Math.*, 25(1):41–50, 1979.
- [36] M. Mulder. The structure of median graphs. *Discret. Math.*, 24(2):197–204, 1978.
- [37] M. Mulder. The interval function of a graph. *Mathematical Centre Tracts, Mathematisch Centrum, Amsterdam*, 1980.
- [38] L. Roditty and V. V. Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proc. of STOC*, pages 515–524, 2013.
- [39] V. Sassone, M. Nielsen, and G. Winskel. A classification of models for concurrency. In *Proc. of CONCUR*, volume 715 of *Lecture Notes in Computer Science*, pages 82–96, 1993.