Subquadratic-time algorithm for the diameter and all eccentricities on median graphs

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Abstract

On sparse graphs, Roditty and Williams [2013] proved that no $O(n^{2-\varepsilon})$-time algorithm achieves an approximation factor smaller than $\frac{3}{2}$ for the diameter problem unless SETH fails. In this article, we solve a longstanding question: can we use the structural properties of median graphs to break this global quadratic barrier?

We propose the first combinatorial algorithm computing exactly all eccentricities of a median graph in truly subquadratic time. Median graphs constitute the family of graphs which is the most studied in metric graph theory because their structure represents many other discrete and geometric concepts, such as CAT(0) cube complexes. Our result generalizes a recent one, stating that there is a linear-time algorithm for computing all eccentricities in median graphs with bounded dimension d, *i.e.* the dimension of the largest induced hypercube (note that 1-dimensional median graphs are exactly the forests). This prerequisite on d is not necessarily anymore to determine all eccentricities in subquadratic time. The execution time of our algorithm is $O(n^{1.6456} \log^{O(1)} n)$.

We provide also some satellite outcomes related to this general result. In particular, restricted to simplex graphs, this algorithm enumerates all eccentricities with a quasilinear running time. Moreover, an algorithm is proposed to compute exactly all reach centralities in time $O(2^{3d} n \log^{O(1)} n)$.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Data structures design and analysis

Keywords and phrases Diameter, Eccentricities, Metric graph theory, Median graphs, Hypercubes.

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Median graphs can be certainly identified as the most important family of graphs in metric graph theory. They are related to numerous areas: universal algebra [4, 15], CAT(0) cube complexes [6, 19], abstract models of concurrency [11, 40], and genetics [9, 10]. Let $d(a,b)$ be the length (*i.e.* the number of edges) of the shortest (a,b)-path for $a,b \in V$ and $I(a,b)$ be the set made up of all vertices u metrically between a and b, *i.e.* $d(a,b) = d(a,u) + d(u,b)$. Median graphs are the graphs such that for any triplet of distinct vertices $x,y,z \in V$, set $I(x,y) \cap I(y,z) \cap I(z,x)$ is a singleton, containing the *median* $m(x,y,z)$ of this triplet.

The purpose of this article is to break the quadratic barrier for the computation time of certain metric parameters on median graphs. In particular, we focus on one of the most fundamental problems in algorithmic graph theory related to distances: the *diameter*. Given an undirected graph $G = (V,E)$, the diameter is the maximum distance $d(u,v)$ for all $u,v \in V$. Two vertices at maximum distance form a *diametral pair*. An even more general problem consists in determining all eccentricities of the graph. The eccentricity $\mathsf{ecc}(v)$ of a vertex v is the maximum length of a shortest path starting from v: $\mathsf{ecc}(v) = \max_{w \in V} d(v,w)$. The diameter is thus the maximum eccentricity.

1.1 State of the art

Executing a *Breadth First Search* (BFS) from each vertex of aninput graph G suffices to obtain its eccentricities in $O(n\,|E|)$, with $n = |V|$. As median graphs are relatively sparse, $|E| \le n \log n$, these multiple BFSs compute all eccentricities in time $O(n^2 \log n)$ for this class of graphs. Very efficient algorithms determining the diameter already exist on other classes of graphs, for example [2, 17, 23]. Many works have also been devoted to approximation algorithms for this parameter. Chechik *et al.* [18] showed that the diameter can be approximated within a factor $\frac{3}{2}$ in time $\tilde{O}(m^{\frac{3}{2}})$ on general graphs. On sparse graphs, it was shown [39] that no $O(n^{2-\varepsilon})$-time algorithm can achieve an approximation factor smaller than $\frac{3}{2}$ for the diameter unless the Strong Exponential Time Hypothesis (SETH) fails.

Median graphs are bipartite and can be isometrically embedded into hypercubes. They admit structural properties, such as the Mulder's convex expansion [36, 37]. They are strongly related to hypercubes retracts [5], Cartesian products and gated amalgams [6], but also Helly hypergraphs [35]. They do not contain induced $K_{2,3}$, otherwise a triplet of vertices would admit at least two medians. The *dimension* d of a median graph G is the dimension of its largest induced hypercube. The value of this parameter is at most $\lfloor \log n \rfloor$ and meets this upper bound when G is an hypercube. Moreover, parameter d takes part in the sparsity of median graphs: $|E| \le dn$.

An important concept related to median graphs is the equivalence relation Θ. This is the reflexive and transitive closure of relation Θ_0, where two edges are in Θ_0 if they are opposite in a common 4-cycle. A Θ-*class* is an equivalence class of Θ. Each Θ-class of a median graph forms a matching cutset, splitting the graph into two convex connected components, called *halfspaces*. The number $q \le n$ of Θ-classes corresponds to the dimension of the hypercube in which the median graph G isometrically embeds. Value q satisfies the Euler-type formula $2n - m - q \le 2$ [31]. A recent LexBFS-based algorithm [13] identifies the Θ-classes in linear time $O(|E|) = O(dn)$.

Two subquadratic-time algorithms have been proposed for the recognition of median graphs. Using convex characterizations of halfspaces, Hagauer *et al.* [26] showed that median graphs can be recognized in $O(n^{\frac{3}{2}} \log n)$. In [28], a bijection between median and triangle-free graphs makes the recognition algorithms for triangle-free graphs work on median ones [3].

Hence, median graphs can be recognized in $O((n \log^2 n)^{1.41})$ using this reduction.

There exist efficient algorithms for some metric parameters on median graphs. For example, the median set and the Wiener index can be determined in $O(|E|)$ [13]. Subfamilies of median graphs have also been studied. There is an algorithm computing the diameter and the radius in linear time for squaregraphs [20]. A more recent contribution introduces a quasilinear time algorithm - running in $O(n \log^{O(1)} n)$ - for the diameter on cube-free median graphs [22], using distance and routing labeling schemes proposed in [21]. Eventually, a linear-time algorithm [14] for the diameter on constant-dimension median graphs was proposed, *i.e.* for median graphs satisfying $d = O(1)$.

The existence of a truly subquadratic-time algorithm for the diameter on all median graphs is open and was recently formulated in [13, 14, 22]. An even more ambitious question can be asked. Can this subquadratic barrier be overpassed for the problem of finding all eccentricities of a median graph ? As the total size of the output is linear and this problem generalizes the diameter one, this question is legitimate. More generally, the question holds for all metric parameters (except the median set and the Wiener index for which a linear-time algorithm was recently designed). In this article, we propose the first subquadratic-time algorithm computing all eccentricities on median graphs.

1.2 Contributions

Our first contribution in this paper is the design of a quasilinear, *i.e.* $O((\log n)^{O(1)} n)$, time algorithm computing the diameter of simplex graphs. A simplex graph $K(G) = (V_K, E_K)$ of a graph G is obtained by considering the induced complete graphs (cliques) of G as vertices V_K. Then, two of these cliques are connected by an edge if they differ by only one element: one is C, the other is $C \cup \{v\}$. These edges form the set E_K. All simplex graphs are median [6, 12]. Moreover, we observe that simplex graphs fulfil an interesting property: they admit a central vertex - representing the empty clique - and every Θ-class has an edge incident to that vertex.

We describe the algorithm in a few words. We observed that the eccentricities of each vertex of a simplex graph could be written as functions of the size of certain sets of pairwise orthogonal Θ-classes (POFs). Based on that property, we order the POFs in function of their size and execute partition refinements. This reveals us a DAG structure of the POFs from which the eccentricity of each vertex can be extracted.

First, this algorithm extends the set of median graphs for which a quasilinear time procedure computing the diameter exists. Indeed, simplex graphs form a sub-class of median graphs containing instances with unbounded dimension d.

- There is a combinatorial algorithm determining the diameter and all eccentricities of simplex graphs in $O((d^3 + \log n)n)$: Theorem 3.1, Section 3.

Second, we remark that this method can be integrated to the algorithm already proposed in [14] to compute all eccentricities of median graphs in time $O(2^{O(d \log d)} n)$. This allows us to decrease this running time. Thanks to this modification, the new algorithm proposed computes all eccentricities of a median graph in $\tilde{O}(2^{2d} n)$, where notation $\tilde{O}$ neglects poly-logarithmic factors. Even if the algorithm stays linear for constant-dimension median graphs, observe that the dependence on d decreases, from a slightly super-exponential function to a simple exponential one.

- There is a combinatorial algorithm determining all eccentricities of median graphs in $\tilde{O}(2^{2d} n)$: Theorem 4.1, Section 4.1.

The second and main contribution in this paper is the design of a subquadratic-time dynamic programming procedure which compute all eccentricities of any median graph. Here, the linear-time simple-exponential-FPT algorithm for all eccentricities presented above plays a crucial role: it is the base case. This framework consists in partitioning recursively the input graph G into the halfspaces of its largest Θ-classes. With our construction, the leaves of this recursive tree are median graphs with dimension at most $\frac{1}{3}\log n$ and we can apply the former linear-time FPT algorithm.

- There is a combinatorial algorithm determining all eccentricities of median graphs in $\tilde{O}(n^{\frac{5}{3}})$: Theorem 4.7, Section 4.2.

We terminate the article with some improvements of the framework we designed. We focus on the computation of all reach centralities [24] in a median graph. The reach centrality of a vertex u is the maximum value $\min\{d(s,u), d(u,t)\}$ over all pairs s,t satisfying $u \in I(s,t)$. A linear-time simple-exponential-FPT algorithm is proposed, as for eccentricities.

- There is a combinatorial algorithm determining all reach centralities of median graphs in $\tilde{O}(2^{3d}n)$: Theorem 5.1, Section 5.1.

Furthermore, we define a new discrete structure on median graphs: the *maximal outgoing POFs* (MOPs), generalizing the POFs used throughout the paper. We propose an alternative procedure to compute all eccentricities, based on the enumeration of MOPs. Furthermore, the MOPs admit an interesting property: for median graphs with “large” d, their number is subquadratic. This provides us with a better subquadratic-time algorithm for the eccentricities using the following win-win approach: either the dimension d of the input graph G is “small” and the linear-time FPT algorithm is executed fast, or the dimension is “large”, then the MOPs can be enumerated fast and our new procedure ensures a smaller runtime.

- There is a combinatorial algorithm determining all eccentricities in $\tilde{O}(n^{\beta})$, where $\beta = 1.6456$: Theorem 5.11, Section 5.2.

All these outcomes put in evidence a relationship between the design of linear-time FPT algorithms and the design of subquadratic-time algorithms determining metric parameters on median graphs. We believe that the ideas proposed to establish all these results represent interesting tools to break the quadratic barrier on other open questions.

1.3 Organization

In Section 2, we remind the definition of median graphs. The well-known properties and concepts related to them are listed, among them Θ-classes, signature, and POFs. Section 3 is utterly dedicated to simplex graphs: we establish some characterizations of these graphs and we present our quasilinear-time algorithm determining their eccentricities. In Section 4, we show how to obtain a linear-time simple-exponential-FPT algorithm for all eccentricities of a median graph, parameterized by the dimension d. Thanks to it, we propose a dynamic programming procedure to reduce the computation of eccentricities of any median graph to the same problem on a collection of median subgraphs of sub-logarithmic dimension. In Section 5, we extend the results obtained so far. We present a linear simple-exponential-FPT algorithm computing all reach centralities of a median graph, parameterized by d. Moreover, we define the notion of MOPs, provide an upper bound of their cardinality and show the impact of this bound on the time complexity on the algorithms proposed earlier. Eventually, we give in Section 6 some directions of research which could follow the contributions of this article. Due to the page limit, some definitions, properties and proofs of Sections 2-5 are put in Appendix.

2 Median graphs

In this section, we recall some notions related to distances in graphs, and more particularly median graphs. Two important tools are presented: the Θ-classes, which are equivalences classes over the edge set, and the *Pairwise Orthogonal Families* (POFs) characterizing Θ-classes belonging to a common hypercube.

2.1 Θ-classes

All graphs $G = (V, E)$ considered in this paper are undirected, unweighted, simple, finite and connected. We denote by $N(u)$ the *open neighborhood* of $u \in V$, *i.e.* the set of vertices adjacent to u in G. We extend it naturally: for any set $A \subseteq V$, the neighborhood $N(A)$ of A is the set of vertices outside A adjacent to some $u \in A$.

Given two vertices $u, v \in V$, let $d(u, v)$ be the *distance* between u and v, *i.e.* the length of the shortest (u, v)-path. The *eccentricity* $\mathsf{ecc}(u)$ of a vertex $u \in V$ is the length of the longest shortest path starting from u. Put formally, $\mathsf{ecc}(u)$ is the maximum value $d(u, v)$ for all $v \in V$: $\mathsf{ecc}(u) = \max_{v \in V} d(u, v)$. The diameter of graph G is the maximum distance between two of its vertices: $\mathsf{diam}(G) = \max_{u \in V} \mathsf{ecc}(u)$.

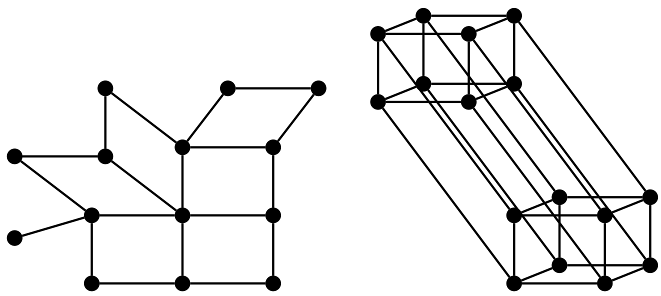
We denote by $I(u, v)$ the *interval* of pair u, v. It contains exactly the vertices which are metrically between u and v: $I(u, v) = \{x \in V : d(u, x) + d(x, v) = d(u, v)\}$. The vertices of $I(u, v)$ are lying on at least one shortest (u, v)-path.

We say that a set $H \subseteq V$ (or the induced subgraph $G[H]$) is *convex* if $I(u, v) \subseteq H$ for any pair $u, v \in H$. Moreover, we say that H is *gated* if any vertex $v \notin H$ admits a *gate* $g_H(v) \in H$, *i.e.* a vertex that belongs to all intervals $I(v, x)$, $x \in H$. For any $x \in H$, we have $d(v, g_H(v)) + d(g_H(v), x) = d(v, x)$. Gated sets are convex by definition.

Given an integer $k \geq 1$, the hypercube of dimension k, Q_k, is a graph representing all the subsets of $\{1, \ldots, k\}$ as the vertex set. An edge connects two subsets if one is included into the other and they differ by only one element. Hypercube Q_2 is a *square* and Q_3 is a 3-cube.

▶ **Definition 2.1** (Median graph). *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 $m(x, y, z)$ called the* median *of x, y, z.*

Observe that certain well-known families of graphs are median: trees, grids, squaregraphs [8], and hypercubes Q_k. Median graphs are bipartite and do not contain an induced $K_{2,3}$ [6, 27, 36]. They can be obtained by Mulder's convex expansion [36, 37] starting from a single vertex.



(a) Tree, $d = 1$ **(b)** Squaregraph, $d = 2$ **(c)** 4-cube, $d = 4$

Figure 1 Examples of median graphs

Now, we define a parameter which has a strong influence on the study of median graphs. The dimension $d = \dim(G)$ of a median graph G is the dimension of the largest hypercube contained in G as an induced subgraph. In other words, G admits Q_d as an induced subgraph, but not Q_{d+1}. Median graphs with $d = 1$ are exactly the trees. Median graphs with $d \leq 2$ are called *cube-free* median graphs.

Figure 1 presents three examples of median graphs. (a) is a tree: $d = 1$. (b) is a cube-free median graph: it has dimension $d = 2$. To be more precise, it is a squaregraph [8], which is a sub-family of cube-free median graphs. The last one (c) is a 4-cube: it has dimension $d = 4$.

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 [13, 26, 28].

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

▶ **Lemma 2.2** (Convex⇔Gated [6, 13])**.** *Any convex subgraph of a median graph is gated.*

To improve readibility, edges $(u, v) \in E$ are sometimes denoted by uv. We remind the notion of Θ-class, which is well explained in [13], 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 $E_1, \ldots, E_q$. Concretely, two edges uv and $u'v'$ belong to the same Θ-class if there is a sequence $uv = u_0v_0, u_1v_1, \ldots, u_rv_r = u'v'$ such that u_iv_i and $u_{i+1}v_{i+1}$ are opposite edges of a square. We denote by $\mathcal{E}$ the set of Θ-classes: $\mathcal{E} = \{E_1, \ldots, E_q\}$. 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 $\lfloor \log n \rfloor$ in general.

▶ **Lemma 2.3** (Θ-classes in linear time [13])**.** *There exists an algorithm which computes the Θ-classes $E_1, \ldots, E_q$ of a median graph in linear time $O(|E|) = O(dn)$.*

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 2.4** (Halfspaces of E_i [13, 26, 37])**.** *For any $1 \leq i \leq q$, the graph G deprived of edges of E_i,* i.e. *$G \backslash E_i = (V, E \backslash E_i)$, 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 $H''_i = W(v, u) = \{x \in V : d(x, v) < d(x, u)\}$.*

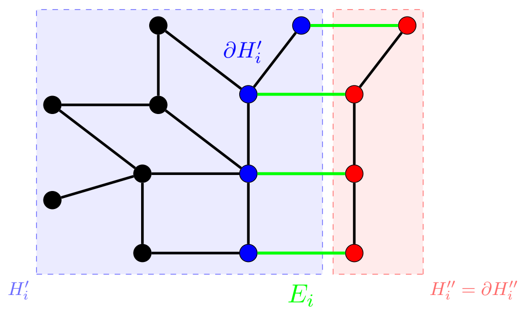


Figure 2 A class E_i with sets $H'_i, H''_i, \partial H'_i, \partial H''_i$

We denote by $\partial H_i'$ the subset of H_i' containing the vertices which are adjacent to a vertex in H_i'': $\partial H_i' = N(H_i'')$ Put differently, set $\partial H_i'$ is made up of vertices of H_i' which are endpoints of edges in E_i. Symmetrically, 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. Figure 2 illustrates the notions of Θ-class, halfspace and boundaries on a small example. In this particular case, an halfspace is equal to its boundary: $\partial H_i'' = H_i''$. The vertices of $\partial H_i'$ are colored in blue.

▶ **Lemma 2.5** (Boundaries [13, 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 too and it belongs to E_j. 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 $\overrightarrow{uv}$ 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. The v_0-orientation is acyclic.

▶ **Lemma 2.6** (Orientation [13]). *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.

2.2 Shortest paths and signature

This subsection is put in Appendix A.1.

2.3 Orthogonal Θ-classes and hypercubes

We present now another important notion on median graphs: *orthogonality*.

▶ **Definition 2.7** (Orthogonal Θ-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 say that E_i and E_j are *parallel* if they are not orthogonal, that is $H_i \subseteq H_j$ for some $H_i \in \{H_i', H_i''\}$, $H_j \in \{H_j', H_j''\}$. We define the sets of pairwise orthogonal Θ-classes.

▶ **Definition 2.8** (Pairwise Orthogonal Family). *We say that a set of classes $X \subseteq \mathcal{E}$ is a* Pairwise Orthogonal Family (POF for short) *if for any pair $E_j, E_h \in X$, we have $E_j \perp E_h$.*

For any induced hypercube of G, its *basis* (resp. *anti-basis*) is the closest vertex (resp. farthest) to v_0 in it. All edges of the hypercube indicent to the basis are outgoing from it. Hypercubes are in bijection with pairs (u, L), where u is a vertex (the basis of the hypercube) and L is a POF outgoing from u (the *signature* of the hypercube).

The full version of this subsection is put in Appendix A.2.

2.4 Maximal POFs

We terminate this preliminary section with a few words on maximal POFs, *i.e.* POFs X such that there is no other POF $Y \supsetneq X$. We highlight a natural bijection between maximal POFs and maximal hypercubes. Its proof is put in Appendix A.3.

▶ **Theorem 2.9** (A.11). *For any maximal induced hypercube, the Θ-classes of its edges form a maximal POF. Conversely, for any maximal POF X, there exists a unique hypercube of signature X.*

3 Simplex graphs

There is a combinatorial algorithm computing the diameter and all eccentricities of simplex graphs in quasilinear time. Properties of simplex graphs, the design of this algorithm and its analysis are put in Appendix B.

▶ **Theorem 3.1** (B.9). *There is a combinatorial algorithm determining all eccentricities of a simplex graph in time $O((d^3 + \log n)n)$.*

4 Subquadratic-time algorithm for all eccentricities on median graphs

This section introduces the design of algorithms computing all eccentricities for the whole class of median graphs (not only simplex graphs). We begin in Section 4.1 with the proposal of a linear-time FPT algorithm, parameterized by the dimension d, running in $2^{O(d)}n$. It is based on some techniques of a paper of the literature [14] which provides a slightly super-exponential time algorithm - running in $2^{O(d\log d)}n$ - for the same problem. We prove that replacing one step of this procedure by the partitioning conceived in Section 3 allows us to decrease the exponential dependence on d.

Thanks to this outcome, in Section 4.2, we are able to design a first subquadratic-time algorithm for all median graphs running in $\tilde{O}(n^{\frac{5}{3}})$.

4.1 Linear FPT algorithm for constant-dimension median graphs

This subsection is dedicated to the construction of an exact algorithm determining all eccentricities on median graphs in $\tilde{O}(2^{2d}n)$. The reader can find it in Appendix C.1.

▶ **Theorem 4.1** (C.16). *There is a combinatorial algorithm computing the list of all eccentricities of a median graph G in time $\tilde{O}(2^{2d}n)$.*

4.2 Tackling the general case

Our new FPT algorithm for computing the list of eccentricities in a median graph has a runtime in $2^{\mathcal{O}(d)}n$, with d being the dimension (Theorem 4.1). This runtime stays subquadratic in n as long as $d < \alpha \cdot \log n$, for some constant $\alpha < 1$. In what follows, we present a simple partitioning scheme for median graphs into convex subgraphs of dimension at most $\alpha \cdot \log n$, for an arbitrary value of $\alpha \le 1$. By doing so, we obtain (in combination with Theorem 4.1) the first known subquadratic-time algorithm for computing all eccentricities in a median graph.

The proof of the Lemmas which are not given in this subsection are put in Appendix C.2. We start with a simple relation between the eccentricity function of a median graph and the respective eccentricity functions of any two complementary halfspaces.

▶ **Lemma 4.2** (C.17). *Let G be a median graph. For every $1 \le i \le q$, let $v \in V(H'_i)$ be arbitrary, and let $v^* \in \partial H''_i$ be its gate. Then, $ecc(v) = \max\{ecc_{H'_i}(v), d(v, v^*) + ecc_{H''_i}(v^*)\}$.*

We will use this above Lemma 4.2 later in our proof in order to compute in linear time the list of eccentricities in a median graph being given the lists of eccentricities in any two complementary halfspaces.

Next, we give simple properties of Θ-classes, to be used in the analysis of our main algorithm in this section (see Lemma 4.6).

▶ **Lemma 4.3** (C.18). *Let H and G be median graphs. If H is an induced subgraph of G then, every Θ-class of H is contained in a Θ-class of G.*

This above Lemma 4.3 can be strenghtened in the special case of *isometric* subgraphs, namely:

▶ **Lemma 4.4** (C.19). *Let H and G be median graphs, and let $E_1, E_2, \ldots, E_q$ denote the Θ-classes of G. If H is an isometric subgraph of G then, the Θ-classes of H are exactly the nonempty subsets among $E_i \cap E(H)$, for $1 \leq i \leq q$.*

An important consequence of Lemma 4.3 is the following relation between the dimension d of a median graph and the cardinality of its Θ-classes.

▶ **Lemma 4.5** (C.20). *Let G be a median graph, and let $D := \max\{|E_i| \mid 1 \leq i \leq q\}$ be the maximum cardinality of a Θ-class of G. Then, $d = \dim(G) \leq \lfloor \log D \rfloor + 1$, and this bound is sharp.*

We are now ready to present our main technical contribution in this section.

▶ **Lemma 4.6.** *If there is an algorithm for computing all eccentricities in an n-vertex median graph of dimension at most d in $\tilde{O}(c^d \cdot n)$ time, then in $\tilde{O}(n^{2-\frac{1}{1+\log c}})$ time we can compute all eccentricities in* any *n-vertex median graph.*

Proof. Let G be an n-vertex median graph. We compute its Θ-classes $E_1, E_2, \ldots, E_q$, that takes linear time (Lemma 2.3). For some parameter D (to be fixed later in the proof), let us assume without loss of generality $E_1, E_2, \ldots, E_p$ to be the subset of all Θ-classes of cardinality $\geq D$, for some $p \leq q$. Note that we have $p \leq |E|/D = \tilde{O}(n/D)$.

We reduce the problem of computing all eccentricities in G to the same problem on every connected component of $G \setminus (E_1 \cup E_2 \cup \ldots \cup E_p)$. More formally, we construct a rooted binary tree[1] T, whose leaves are labelled with convex subgraphs of G. Initially, T is reduced to a single node with label equal to G. Then, for $i = 1 \ldots p$, we further refine this tree so that, at the end of any step i, its leaves are labelled with the connected components of $G \setminus (E_1 \cup E_2 \cup \ldots \cup E_i)$. An example of T is shown in Figure 3 with $D = 3$ and two Θ-classes reaching this cardinality bound.

For that, we proceed as follows. We consider all leaves of T whose label H satisfies $E(H) \cap E_i \neq \emptyset$. By Lemma 4.4, $E(H) \cap E_i$ is a Θ-class of H. We make of both corresponding halfspaces of H the left and right children of H in T. Recall that the leaves of T at this step i are the connected components of $G \setminus (E_1 \cup E_2 \cup \ldots E_{i-1})$, and in particular that they form a partition of $V(G)$. Therefore, each step takes linear time by reduction to computing the connected components in vertex-disjoint subgraphs of G. Overall, the total time for constructing the tree T is in $O(pm) = \tilde{O}(n^2/D)$.

Then, we compute the list of eccentricities for all the subgraphs labelling a node, by dynamic programming on T. In particular, doing so we compute the list of eccentricities for G because it is the label of the root. There are two cases:

- If H labels a leaf (base case) then, we claim that we have $\dim(H) \leq \lfloor \log D \rfloor + 1$. Indeed, by Lemma 4.3, every Θ-class of H is contained in a Θ-class of G. Since we removed all Θ-classes of G with at least D edges, the claim now follows from Lemma 4.5. In particular, we can compute the list of all eccentricities for H in $\tilde{O}(c^{\lfloor \log D \rfloor + 1}|V(H)|) = \tilde{O}(D^{\log c}|V(H)|)$ time. Recall that the leaves of T partition $V(G)$, and therefore, the total runtime for computing the list of eccentricities for the leaves is in $\tilde{O}(D^{\log c} n)$.

[1] This tree T is independent to the one built in Appendix B.2

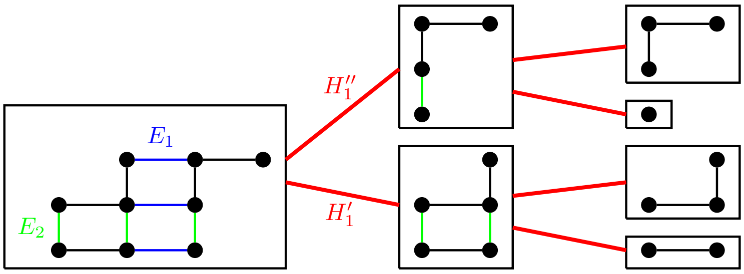


Figure 3 An example of tree T associated with a graph G for $D = 3$: here, $p = 2$.

- From now on, let us assume H to label an internal node of T (inductive case). Let H_i', H_i'' be its children nodes, obtained from the removal of $E(H) \cap E_i$ for some $1 \leq i \leq p$. – For convenience, we will say later in the proof that H is a i-node. – Recall that $E(H) \cap E_i$ is a Θ-class of H. In particular, H_i', H_i'' are gated subgraphs. By Lemma 4.2, we can compute in $O(|V(H_i')|)$ time the eccentricities in H of all vertices in H_i' if we are given as input: the list of eccentricities in H_i', the list of eccentricities in H_i'', and for every $v \in V(H_i')$ its gate $v^* \in \partial H_i''$ and the distance $d(v, v^*)$. The respective lists of eccentricities for H_i' and H_i'' were pre-computed by dynamic programming on T. Furthermore, we can compute the gate v^* and $d(v, v^*)$ for every vertex $v \in V(H_i')$, in total $\tilde{O}(|V(H)|)$ time, by using a modified BFS rooted at H_i'' (we refer to [21, Lemma 17] for a detailed description of this standard procedure). Overall (by proceeding the same way for H_i'' as for H_i') we can compute the list of eccentricities for H in $\tilde{O}(|V(H)|)$ time. This is in total $\tilde{O}(n)$ time for the i-nodes (*i.e.*, because they were leaves of T at step i, and therefore, they are vertex-disjoint), and so, in total $\tilde{O}(pn) = \tilde{O}(n^2/D)$ time for all the internal nodes.

The total runtime for our algorithm is in $\tilde{O}(n^2/D + D^{\log c} n)$, that is optimized for $D = n^{\frac{1}{\log c + 1}}$. ◀

▶ **Theorem 4.7.** *There is an $\tilde{O}(n^{5/3})$-time algorithm for computing all eccentricities in median graphs.*

Proof. This result directly follows from the combination of Theorem 4.1 with Lemma 4.6 (applied for $c = 4$). ◀

Observe that the design of a linear-time FPT algorithm for eccentricities in $\tilde{O}(c^d n)$ with $c < 4$ would imply a lower subquadratic constant for this problem.

5 Generalization and improvements

In this section, we discuss some consequences and possible improvements of the algorithms established in Section 4.

First, we focus on another metric parameter called *reach centrality*. We show the existence of an exact algorithm for all reach centralities in $\tilde{O}(2^{3d} n)$ on median graphs.

Second, we propose a discrete structure strongly related to both POFs and hypercubes but slightly different to them: *Maximal Outgoing POFs*, also called MOPs. We introduce a different way to compute all eccentricities of a median graph based on this structure. This yields a second subquadratic-time algorithm with a smaller exponent.

5.1 Reach centrality

The content of this subsection is put in Appendix D.1. It presents the following result.

▶ **Theorem 5.1** (D.2). *There is a combinatorial algorithm computing all reach centralities $RC(u)$ of a median graph in $\tilde{O}(2^{3d}n)$.*

5.2 MOP structure

In this subsection, we introduce a new discrete structure for median graphs, called Maximal Outgoing POFs (MOPs). We present another way to determine all eccentricities based on the enumeration of MOPs. Thanks to this result, we obtain an improvement of Theorem 4.7 via a win-win approach. When $d \leq a^* \log n$ (value $a^* < 1$ will be determined in the proof), we can apply Theorem 4.1. Otherwise, when $d > a^* \log n$, we show that G admits a subquadratic number of MOPs and the eccentricities can be computed more efficiently than in Section 4.1. The reader will found the missing proofs in Appendix D.2.

5.2.1 Definition and relationship with eccentricities

We remind that a pair made up of a vertex u and a POF L outgoing from this vertex can be seen as a hypercube (of basis u and signature L, which is unique). The MOPs are defined to highlight certain hypercubes which satisfy a maximality property.

▶ **Definition 5.2** (Maximal Outgoing POFs). *Pair (u, L) is a MOP if L is outgoing from u and there is no other $L' \supsetneq L$ outgoing from u.*

On one hand, we can associate with each MOP (u, L) the unique hypercube with basis u and signature L. However, there are some hypercubes such that their pair basis-signature is not a MOP. As a trivial example, consider the square C_4 with Θ-classes E_1, E_2. The two edges which are incident to v_0 are hypercubes of dimension 1, $(v_0, \{E_1\})$ and $(v_0, \{E_2\})$, but are not MOPs since the POF $\{E_1, E_2\}$ is outgoing from v_0 and maximal.

On the other hand, there is an interesting relationship between MOPs and maximal POFs. We remind that maximal POFs are in bijection with maximal induced hypercubes (Theorem 2.9). Thus, a maximal POF is a MOP if we consider the pair basis-signature of the maximal hypercube representing it. Conversely, MOPs are not necessarily signed with maximal POFs. Let us consider the same trivial example C_4: the two edges which are not incident to v_0 are MOPs but do not form a maximal hypercube. In brief, MOPs represent some intermediary discrete structure between hypercubes and maximal hypercubes (or maximal POFs).

The execution time of the algorithms (Theorems 4.1 and 4.7) we designed to determine the eccentricities of median graphs depend on the listing of all pairs (L, R) of POFs such that L (resp. R) is outgoing from (resp. ingoing into) vertex u and $R \cup \{E_i\}$ is not a POF for any $E_i \in L$ (see Appendix C for details). We show how the MOPs offer an alternative to this "brute force" enumeration. In fact, we can determine all eccentricities by listing only these pairs (L, R) for which (u, L) is a MOP (instead of being an hypercube).

▶ **Theorem 5.3** (D.3). *Assume a median graph G has at most $\tilde{O}(f(d, n)n)$ MOPs, $f(d, n) = o(2^d)$. There is a combinatorial algorithm computing all its eccentricities in $\tilde{O}(2^d f(d, n)n)$.*

5.2.2 Cardinality of MOPs

Our objective is now to express the cardinality of MOPs in function of n and d in order to apply Theorem 5.3 and improve the subquadratic execution time established in Theorem 4.7. To do so, we introduce a relationship between MOPs and the subsets of maximal POFs.

▶ **Definition 5.4.** *Let p be the application which, given $u \in V$ and a POF L outgoing from u, returns pair (L, L^*), where L^* is the POF of Θ-classes ingoing into the anti-basis of (u, L).*

If we restrict application p to MOPs, it returns a pair made up of a maximal POF and one of its subsets.

▶ **Lemma 5.5** (D.4). *Let (u, L) be a MOP, $p(u, L) = (L, L^*)$. Then, L^* is a maximal POF.*

Maximal POFs can be interpreted in the crossing graph $G^\#$ of G, whose vertex set contains the Θ-classes of G and two of them are adjacent if they are orthogonal (see [29] or Definition B.3 in Appendix B). A POF of G is exactly a clique of $G^\#$ and a maximal POF corresponds to a maximal clique of $G^\#$. We provide an upper bound of the number of MOPs of G depending on the maximal cliques of its crossing graph.

▶ **Corollary 5.6** (D.5). *Let $G^\#$ be the crossing graph of G and $\mathcal{C}^\#_{max}$ be the set of maximal cliques of $G^\#$. The number of MOPs in G is at most $\sum\limits_{C \in \mathcal{C}^\#_{max}} 2^{|C|}$.*

▶ **Definition 5.7** (Maximal clique ratio). *The maximal clique ratio $r(H)$ of a graph H is the quotient between the sum of the number of subsets of each maximal clique of H by the number of cliques of H. Formally,*

$$r(H) = \frac{R\left[H\right]}{N\left[H\right]} = \frac{\sum\limits_{C \in \mathcal{C}_{max}(H)} 2^C}{|\mathcal{C}(H)|}$$

The *clique number* of a graph is the size of its maximum clique. The dimension d of G is also the clique number of $G^\#$. Complete multipartite graphs (with clique number d) are the graphs whose vertex set can be partitioned into independent sets A_i, $1 \leq i \leq d$, and any pair of vertices belonging to a different set form an edge.

We begin with the proof that the complete multipartite graphs maximize the ratio $r(H)$. Next, we show that the more complete multipartite graph are balanced, the largest $r(H)$ is.

The complete multipartite graphs are exactly the graphs fulfilling the following property: for any non-adjacent vertices u, v, $N(u) = N(v)$. Let $\mathrm{Trp}(H)$ be the number of triplets (u, v, w) of vertices of H such that $uv \notin E$, $uw \notin E$, but $vw \in E$. An equivalent way to characterize complete multipartite graphs is $\mathrm{Trp}(H) = 0$. In other words, any graph which is not complete multipartite verify $\mathrm{Trp}(H) > 0$.

▶ **Theorem 5.8** (D.6). *Let H be a graph with $|V(H)| = q$, clique number at most d, which maximizes $r(H)$. If H is not complete multipartite, there is another graph H' with $|V(H')| = q$, clique number at most d, such that $r(H') = r(H)$ and $\mathrm{Trp}(H') < \mathrm{Trp}(H)$.*

By successive applications of this result, for any graph H of clique number at most d maximizing $r(H)$, there exists a complete multipartite graph H' with the same ratio and clique number at most d. Turán graphs $T(q, d)$ are the most balanced complete multipartite graphs with q vertices and clique number d. The size of two of its independent sets differ of at most one. Now, the objective is to prove that, among complete multipartite graphs, Turán graphs maximize the maximal clique ratio.

▶ **Theorem 5.9** (D.7). *Turán graphs $T(q,d)$ maximize the maximal clique ratio for graphs with q vertices and clique number d.*

Naturally, we use the maximal clique ratio of Turán graphs to deduce an upper bound for the number of MOPs of any median graph G.

▶ **Corollary 5.10** (D.8). *The number of MOPs in a median graph is $O(f(d,n)n)$, where $f(d,n) = \left(2.\frac{2^{\frac{\log n}{d}}-1}{2^{\frac{\log n}{d}}}\right)^d$.*

By observing the expression of function $f(d,n)$, we see that the larger the dimension d, the smaller the number of MOPs. The following result comes from the win-win approach we announced earlier.

▶ **Theorem 5.11.** *There is a combinatorial algorithm determining all eccentricities in time $\tilde{O}(n^{1.6456})$ on median graphs.*

Proof. Let $a = \frac{d}{\log n}$ and we define two functions: $f(x) = 2.\frac{2^{\frac{1}{x}}-1}{2^{\frac{1}{x}}}$ and $g(x) = 2 - \frac{1}{1+\log(2f(x))}$.

One one hand, according to Theorem 4.1, there is a combinatorial algorithm determining all eccentricities in $\tilde{O}(2^{2d}n) = \tilde{O}(n^{1+2a})$. On the other hand, according to Theorem 5.3 and Corollary 5.10, we can also compute all eccentricities in time $\tilde{O}(2^d(f(a))^d n)$. Using the reduction scheme of Theorem 4.6, we obtain them in subquadratic time $\tilde{O}(n^{g(a)})$.

To obtain the best runtime possible, we have to minimize the subquadratic constant $h(a) = \max\{1+2a, g(a)\}$. Function h admits a unique minimum for $0 < a \leq 1$, reached for a certain a^* we can approximate by $0.3327 \leq a^* \leq 0.3328$. This gives $h(a^*) \simeq 1.6456$.

We describe the combinatorial algorithm computing all eccentricities in $\tilde{O}(n^{h(a^*)})$. We determine d with a BFS from v_0 as it is equal to the maximum number of edges incoming into a vertex of G (Lemma A.8 of Appendix A.2). If $\frac{d}{\log n} \leq a^*$, then apply the linear FPT algorithm evoked in Theorem 4.1. Otherwise, if $\frac{d}{\log n} > a^*$, then use the enumeration of MOPs (Theorem 5.3) and apply the reduction scheme proposed in Lemma 4.6. ◀

6 Conclusion

As a natural extension of this work, the question of designing a linear-time or quasilinear-time algorithm to compute the diameter and all eccentricities of median graphs is now open. With the recursive splitting procedure of Lemma 4.6, unfortunately, the best execution time we could obtain is $\tilde{O}(n^{\frac{3}{2}})$. Reaching this bound could represent a first reasonable objective: it would "suffice" to propose a FPT combinatorial algorithm which computes all eccentricities in $\tilde{O}(2^d n)$ in order to obtain such time complexity. We see the MOP-approach as a gateway to identify such a procedure.

Another - certainly easier - objective after this work is to adapt the recursive splitting of Lemma 4.6 for reach centralities. We tried to define a weighted version of the reach centralities problem in order to fit them to the halfspace separation, but this task seems to be not so easy. Our hope is to obtain a subquadratic-time algorithm computing reach centralities in median graphs.

Eventually, we note two lines of research on which this paper could have some influence: (i) the study of efficient algorithms for the computation of other metric parameters on median graphs (perhaps, the *betweeness centrality* [1]) and (ii) the design of subquadratic-time algorithms for the diameter and all eccentricities on larger families of graphs (*almost-median* or *semi-median* graphs [16, 32] for example).

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. H. Bandelt. Retracts of hypercubes. *Journal of Graph Theory*, 8(4):501–510, 1984.
6. H. Bandelt and V. Chepoi. Metric graph theory and geometry: a survey. *Contemp. Math.*, 453:49–86, 2008.
7. H. Bandelt, V. Chepoi, A. W. M. Dress, and J. H. Koolen. Combinatorics of lopsided sets. *Eur. J. Comb.*, 27(5):669–689, 2006.
8. H. Bandelt, V. Chepoi, and D. Eppstein. Combinatorics and geometry of finite and infinite squaregraphs. *SIAM J. Discret. Math.*, 24(4):1399–1440, 2010.
9. 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.
10. 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.
11. J. Barthélemy and J. Constantin. Median graphs, parallelism and posets. *Discret. Math.*, 111(1-3):49–63, 1993.
12. J. Barthélemy, B. Leclerc, and B. Monjardet. On the use of ordered sets in problems of comparison and consensus of classifications. *Journal of Classification*, 3:187–224, 1986.
13. 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.
14. P. Bergé and M. Habib. Diameter, radius and all eccentricities in linear time for constant-dimension median graphs. In *Proc. of LAGOS*, 2021.
15. G. Birkhoff and S. A. Kiss. A ternary operation in distributive lattices. *Bull. Amer. Math. Soc.*, 53:745–752, 1947.
16. B. Bresar. Characterizing almost-median graphs. *Eur. J. Comb.*, 28(3):916–920, 2007.
17. S. Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. In *Proc. of SODA*, pages 2143–2152, 2017.
18. 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.
19. V. Chepoi. Graphs of some CAT(0) complexes. *Adv. Appl. Math.*, 24(2):125–179, 2000.
20. 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.
21. 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.
22. G. Ducoffe. Isometric embeddings in trees and their use in distance problems. In *Proc. of MFCS*, volume 202 of *LIPIcs*, pages 43:1–43:16, 2021.
23. 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.
24. R. J. Gutman. Reach-based routing: A new approach to shortest path algorithms optimized for road networks. In *Proc. of ALENEX/ANALC*, pages 100–111, 2004.
25. M. Habib, C. Paul, and L. Viennot. Partition refinement techniques: An interesting algorithmic tool kit. *Int. J. Found. Comput. Sci.*, 10(2):147–170, 1999.
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 W. Imrich, S. Klavzar, and H. M. Mulder. Median graphs and triangle-free graphs. *SIAM J. Discret. Math.*, 12(1):111–118, 1999.

29 S. Klavzar and M. Kovse. Induced cycles in crossing graphs of median graphs. *Discret. Math.*, 309(23-24):6585–6589, 2009.

30 S. Klavzar and H. M. Mulder. Partial cubes and crossing graphs. *SIAM J. Discret. Math.*, 15(2):235–251, 2002.

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 S. Klavzar and S. V. Shpectorov. Characterizing almost-median graphs II. *Discret. Math.*, 312(2):462–464, 2012.

33 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.

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 R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.

39 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.

40 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.

41 Peter M Winkler. Isometric embedding in products of complete graphs. *Discrete Applied Mathematics*, 7(2):221–225, 1984.

A Appendix for Section 2

A.1 Shortest paths and signature

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

▶ **Definition A.1** (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 $G\backslash E_i$. In other words, u and v are in different halfspaces of E_i.*

The signature of two vertices provide us with the composition of any shortest (u, v)-path. Indeed, all shortest (u, v)-paths contain exactly one edge for each class in $\sigma_{u,v}$.

▶ **Lemma A.2** ([14]). *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. Conversely, a path containing at most one edge of each Θ-class is a shortest path between its departure and its arrival.*

This result is a direct consequence of the convexity of halfspaces. By contradiction, a shortest path that would pass through two edges of some Θ-class E_i would escape temporarily from an halfspace, say w.l.o.g H'_i, which is convex (Lemma 2.4).

Definition A.1 can be generalized: given a set of edges $B \subseteq E$, its signature is the set of Θ-classes represented in that set: $\{E_i : uv \in E_i \cap B\}$. 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: the cardinality of the signature is thus equal to the dimension of the hypercube.

A.2 Orthogonal Θ-classes and hypercubes

We present now another important notion on median graphs: *orthogonality*. In [33], 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. Another definition was proven equivalent to this one.

▶ **Definition A.3** (Orthogonal Θ-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 $ux, vy \in E_j$.*

Indeed, classes E_i and E_j are orthogonal if and only if the splits produced by their halfspaces are incompatible.

▶ **Lemma A.4** (Orthogonal⇔Incompatible [14]). *Given two Θ-classes E_i and E_j of a median graph G, the following statements are equivalent:*

- *Classes E_i and E_j are orthogonal,*
- *Splits $\{H'_i, H''_i\}$ and $\{H'_j, H''_j\}$ are incompatible,*
- *The four sets $\partial H'_i \cap \partial H'_j$, $\partial H''_i \cap \partial H'_j$, $\partial H'_i \cap \partial H''_j$, and $\partial H''_i \cap \partial H''_j$ are nonempty.*

We say that E_i and E_j are *parallel* if they are not orthogonal, that is $H_i \subseteq H_j$ for some $H_i \in \{H'_i, H''_i\}$ and $H_j \in \{H'_j, H''_j\}$.

We pursue with a property on orthogonal Θ-classes: if two edges of two orthogonal classes E_i and E_j are incident, they belong to a common square.

▶ **Lemma A.5** (Squares [11, 14]). *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.*

Pairwise orthogonal families. We focus on set of Θ-classes which are pairwise orthogonal.

▶ **Definition A.6** (Pairwise Orthogonal Family). *We say that a set of classes $X \subseteq \mathcal{E}$ is a* Pairwise Orthogonal Family (POF for short) *if for any pair $E_j, E_h \in X$, we have $E_j \perp E_h$.*

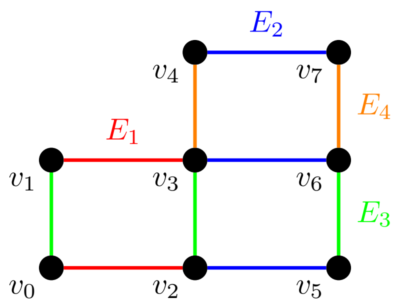
The empty set is considered as a POF. We denote by $\mathcal{L}$ the set of POFs of the median graph G. The notion of POF is strongly related to the induced hypercubes in median graphs. First, observe that all Θ-classes of a median graph form a POF if and only if the graph is an hypercube of dimension $\log n$ [33, 34]. Secondly, the next lemma precises the relationship between POFs and hypercubes.

▶ **Lemma A.7** (POFs adjacent to a vertex [14]). *Let X be a POF, $v \in V$, and assume that for each $E_i \in X$, 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.*

There is a natural bijection between the vertices of a median graph and the POFs. The next lemma exhibits this relationship.

▶ **Lemma A.8** (POFs and hypercubes [7, 9, 33]). *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 $\mathcal{E}^-(v)$. Hence, $|\mathcal{E}^-(v)| = |N^-(v)| \leq d$.*
- *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: $n = |\mathcal{L}|$.*



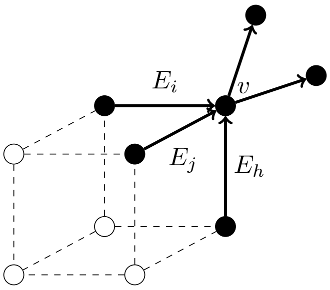
Vertex	v_0	v_1	v_2	v_3
POF	$\emptyset$	$\{E_3\}$	$\{E_1\}$	$\{E_1, E_3\}$

Vertex	v_4	v_5	v_6	v_7
POF	$\{E_4\}$	$\{E_2\}$	$\{E_2, E_3\}$	$\{E_2, E_4\}$

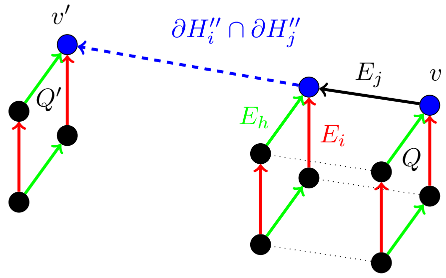
Figure 4 Illustration of the bijection between V and the set of POFs.

An example is given in Figure 4 with a small median graph of dimension $d = 2$. v_0 is the canonical basepoint and edges are colored according to their Θ-class. For example, $v_1v_3 \in E_1$. We associate with any POF X of G the vertex v_X satisfying $\mathcal{E}^-(v_X) = X$ with the v_0-orientation. Obviously, the empty POF is associated with v_0 which has no incoming edges.

A straightforward consequence of this bijection is that parameter q, the number of Θ-classes, is less than the number of vertices n. But it can be used less trivially to enumerate the POFs of a median graph in linear time [9, 33]. 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 from (resp. ingoing into) 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 A.8 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. This observation implies that the number of POFs is less than the number of hypercubes in G. Moreover, the hypercube Q_X is the closest-to-v_0 hypercube formed with the classes in X. Figure 5a 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 $\mathcal{E}^-(v)$.



(a) The hypercube "induced" by the edges ingoing into a vertex (its antibasis).



(b) A POF signing at least two hypercubes Q and Q' is not maximal.

Figure 5 Properties of POFs

Number of hypercubes. 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 [9, 33], we have:

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

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

▶ **Lemma A.9** (Number of hypercubes). *$\alpha(G) \leq 2^d n$.*

Value $\alpha(G)$ consider all hypercubes, in particular those of dimension 0, *i.e.* vertices. From now on, the word "hypercube" refers to the hypercubes of dimension at least one.

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 A.8). 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 $\widehat{N}$ (its signature).

As a consequence, a simple graph search as BFS enables us to enumerate the hypercubes in G in time $O(d2^d n)$.

▶ **Lemma A.10** (Enumeration of hypercubes [14]). *We can enumerate all triplets $(v, u, \widehat{\mathcal{E}})$, where v is the anti-basis of an hypercube Q, u its basis, and $\widehat{\mathcal{E}}$ the signature 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 $(v, u, \widehat{\mathcal{E}})$ containing v appears before any triplet $(v', u', \widehat{\mathcal{E}}')$ containing v'.*

The enumeration of hypercubes is thus executed in linear time for median graphs with constant dimension. In summary, given any median graph, one can compute the set of Θ-classes and their orthogonality relationship (for each E_i, the set of Θ-classes orthogonal to E_i) in linear time, and the set of hypercubes with its basis, anti-basis and signature in $\tilde{O}(2^d n)$.

A.3 Maximal POFs

▶ **Theorem A.11** (Maximal POFs and hypercubes)**.** *For any maximal induced hypercube, the Θ-classes of its edges form a maximal POF. Conversely, for any maximal POF X, there exists a unique hypercube of signature X. Its anti-basis is the vertex v such that $\mathcal{E}^-(v) = X$.*

Proof. Let Q be a maximal hypercube and X_Q its signature. We begin with the proof that X_Q is a maximal POF. Assume that $Y \subsetneq X_Q$. As Y is a POF, there is a vertex $v_Y \in V$ satisfying $\mathcal{E}^-(v_Y) = Y$. Similarly, we denote by v_X the vertex such that $\mathcal{E}^-(v_X) = X_Q$. Both v_X and v_Y belong to the boundary of any Θ-class of X_Q (the one which is the farthest from v_0). In brief,

$$v_X, v_Y \in \bigcap_{E_i \in X_Q} \partial H''_i \tag{2}$$

As every $\partial H''_i$ is gated, then the intersection written in Eq. (2) is convex/gated too. Thus, the shortest (v_X, v_Y)-path is entirely contained in set $\bigcap_{X_Q} \partial H''_i$. Let (v_X, z) be the first edge of this path and E_j the Θ-class of this edge. Each Θ-class form an isomorphism between its two boundaries (Lemma 2.5): as $z \in \bigcap_{X_Q} \partial H''_i$, there is an hypercube isomorphic to Q in the boundary of E_j containing z. Therefore, there is an hypercube of dimension $|X_Q| + 1$ containing all vertices of Q. This yields a contradiction as Q is supposed to be maximal.

Conversely, let Q be an hypercube and assume that its signature X_Q is a maximal POF. We suppose that there is a second hypercube $Q' \neq Q$ such that $X_Q = X_{Q'}$. Then, set $\bigcap_{X_Q} \partial H''_i$ contains at least two elements: the anti-bases of hypercubes Q and Q'. Using the same argument as above, we can put in evidence an edge with two endpoints in $\bigcap_{X_Q} \partial H''_i$. The Θ-class of this edge is thus orthogonal of any class E_i of X_Q which defines an isomorphism between $\partial H'_i$ and $\partial H''_i$. Consequently, we obtain a POF superset of X_Q, a contradiction.

For any POF X, there is at least one hypercube with signature X such that its anti-basis v verifies $\mathcal{E}^-(v) = X$ according to Lemma A.8. In summary, any maximal POF X can be associated with a unique hypercube of signature X. ◀

Figure 5b illustrates this proof with two squares Q and Q' with the same signature $\{E_i, E_h\}$. One can observe the appearance of a hypercube of larger dimension containing Q, giving evidence of the non-maximality of X_Q.

The number of maximal hypercubes in a median graph is thus equal to the number of maximal POFs, which is itself at most linear in the number of vertices.

B Appendix for Section 3

We begin with some properties of simplex graphs and then describe the quasilinear-time algorithm.

B.1 Simplex and crossing graphs: cliques, diameter and opposites

Given any undirected graph G, the vertices of the simplex graph $K(G)$ associated to G represent the induced cliques (not necessarily maximal) of G. Two of these cliques are connected by an edge if they differ by exactly one element.

▶ **Definition B.1** (Simplex graphs [6]). *The simplex graph* $K(G) = (V_K, E_K)$ *of* $G = (V, E)$ *is made up of the vertex set* $V_K = \{C \subseteq V : C \text{ induced complete graph of } G\}$ *and the edge set* $E_K =$
$\{(C, C') : C, C' \in V_K, C \subsetneq C', |C'| - |C| = 1\}$.

Observe that the empty clique of G also corresponds to a vertex $v_\emptyset$ in $K(G)$. Therefore $v_\emptyset$ has degree $n = |V|$ in $K(G)$. Figure 6 shows, as an example, the simplex graph of a C_5. Labels indicate the correspondence between vertices and cliques. We note that simplex graphs of cycles are cogwheels, *i.e.* wheels with a subdivision on each external edge.

The simplex graph of a n-complete graph is an hypercube of dimension n. More generally, simplex graphs are median [6, 12]. We will see later than certain median graphs are not simplex graphs. As a subfamily of median graphs, the design of a subquadratic-time algorithm for the eccentricities on simplex graphs is of interest. Moreover, the dimension d of simplex graphs can be unbounded because hypercubes belong to this class of graphs (their dimension is logarithmic in the vertex set size).

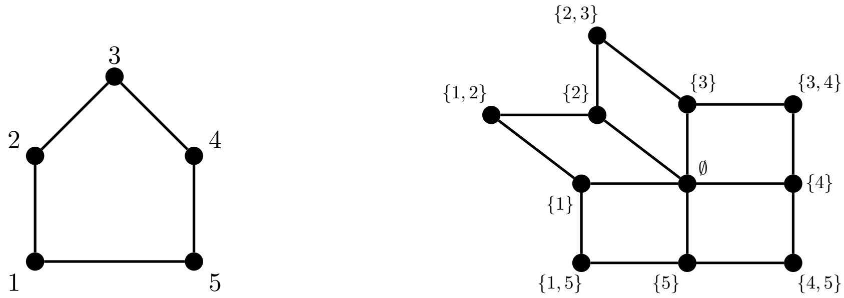


Figure 6 A cycle C_5 and its simplex graph $K(C_5)$

Simplex graphs can be characterized as particular median graphs.

▶ **Theorem B.2.** *Let G be a median graph. The following statements are equivalent:*

(1) *G is a simplex graph.*

(2) *There is a vertex $v_0 \in V(G)$ such that each Θ-class of G is adjacent to v_0,* i.e. $\forall 1 \le i \le q, \exists v_i \in V(G), v_0 v_i \in E_i$.

(3) *There is a vertex $v_0 \in V(G)$ contained in any maximal hypercube of G.*

Proof. (2)⇔(3). Assume each Θ-class is adjacent to v_0. For each POF X, there is an hypercube with signature X and containing v_0 (Lemma A.7). For each maximal hypercube, its signature (the set of Θ-classes it contains) is a maximal POF and, moreover, no other maximal hypercube has the same signature (Theorem A.11). For this reason, each maximal hypercube necessarily contains v_0. Conversely, if any maximal hypercube contains v_0, as each Θ-class belongs to at least one maximal POF, then each Θ-class is necessarily adjacent to v_0.

(2)⇒(1). We consider a median graph G such that all Θ-classes are adjacent to v_0. Our objective is to prove that there exists G' such that $G = K(G')$. Let G' be the graph where its vertices represent the Θ-classes of G and two of them are connected by an edge if the

Θ-classes are orthogonal. In this way, every clique of G' (even the empty one) corresponds to a POF of G. For any POF X of G, its Θ-classes are adjacent to v_0, so there exists an hypercube containing v_0 with signature X, according to Lemma A.7. Moreover, given a POF X, this hypercube is unique. Its anti-basis (opposite of v_0) thus represents the clique X in G'. Conversely, according to the v_0-orientation, each vertex admits its own set of incoming Θ-classes which forms a POF (Lemma A.8). Therefore, we can associate to each vertex of G a clique of G'.

Then, two vertices u, v of G must be adjacent if and only if $\mathcal{E}^-(v) = \mathcal{E}^-(u) \cup \{E_i\}$, for some class E_i. On one hand, suppose $uv \in E_i$. Assume that there exists $E_j \in \mathcal{E}^-(u) \backslash \mathcal{E}^-(v)$. Then, E_j is parallel to E_i, otherwise $E_j \in \mathcal{E}^-(v)$ (Lemma A.5). This is a contradiction: as $H_i'' \subsetneq H_j''$, E_i cannot be adjacent to v_0. Moreover, if $E_h \in \mathcal{E}^-(v) \backslash \mathcal{E}^-(u)$, $h \neq i$, then E_h is necessarily ingoing into u because of Lemma A.5, as E_h and E_i are orthogonal. So, $\mathcal{E}^-(v) \backslash \mathcal{E}^-(u) = \{E_i\}$. On the other hand, if $\mathcal{E}^-(v) = \mathcal{E}^-(u) \cup \{E_i\}$, let v' be the vertex such that $v'v \in E_i$. Again, Lemma A.5 implies that all Θ-classes ingoing into v are also ingoing into v', so $\mathcal{E}^-(u) \subseteq \mathcal{E}^-(v')$. Furthermore, if there is some $E_j \notin \mathcal{E}^-(u)$ ingoing into v', then it should be parallel to E_i, a contradiction. Hence, $v' = u$.

(1)⇒(2). Assume that $G = K(G')$: we denote by $v_\emptyset$ the vertex representing the empty clique. By contradiction, we suppose that there exists a Θ-class which is not adjacent to $v_\emptyset$: we denote it by E_1. We consider the $v_\emptyset$-orientation of the graph. As $\{E_1\}$ is a POF, there is necessarily one vertex v_1 with only one incoming edge belonging to E_1. As $v_1 \neq v_\emptyset$, vertex v_1 represents a clique of G' of size at least 1. If v_1 represents a clique of size exactly 1, then it is adjacent to $v_\emptyset$ because only one element differ between the cliques represented by both $v_\emptyset$ and v_1, which is a contradiction. If v_1 represents a clique of size at least 2, then it must have at least two incoming edges. another contradiction. In summary, if $G = K(G')$, each Θ-class of G is adjacent to $v_\emptyset$. ◀

In this section only, on simplex graphs, the canonical basepoint v_0 is not selected arbitrarily. We fix v_0 as a vertex adjacent to all Θ-classes, as put in evidence by Theorem B.2. We call v_0 the *central vertex* of the simplex graph.

What Theorem B.2 says is that any simplex graph can be seen as a set of maximal POFs (or hypercubes) that are "medianly" assembled. Indeed, one cannot define a simplex graph given any collection of sets - excluding subsets - representing maximal POFs. Consider as an example the collection $\{\{E_1, E_2\}, \{E_2, E_3\}, \{E_3, E_1\}\}$: it would produce a simplex graph with 3 squares with basis v_0 and sharing pairwise an edge. This graph is Q_3^- (the 3-cube minus a vertex) and is not median. The collection implies that E_1, E_2, E_3 are pairwise orthogonal, so $\{E_1, E_2, E_3\}$ should be the maximal POF here.

The most obvious example of median graph which is not simplex is certainly the path P_4. Indeed, it has three Θ-classes which are all pairwise parallel. For any vertex of P_4, there exists a Θ-class which is not adjacent to it.

The proof (2)⇒(1) reveals the reverse application of K.

▶ **Definition B.3** (Crossing graphs [29, 30]). *Let G be a median graph. Its crossing graph $G^\#$ is the graph obtained by considering Θ-classes as its vertices and such that two Θ-classes are adjacent if they are orthogonal.*

Restricted to simplex graphs, this transformation is the reverse of K: indeed, as stated in [30], $G = K(G)^\#$. The clique number of $G^\#$ is exactly the dimension of median graph G. For example, the crossing graph of a cube-free median graph contains no triangle. Each simplex graph admits a central vertex (v_0 in Theorem B.2) which represents the empty clique of $G^\#$.

Now, we focus on the problem of determining a diametral pair of a simplex graph G and more generally all eccentricities. Observe that the distance between the central vertex v_0 and any vertex u of G can be deduced directly from the edges incoming into u. We state that $\sigma_{v_0,u} = \mathcal{E}^-(u)$. This is a consequence of Theorem B.2: all Θ-classes of $\mathcal{E}^-(u)$ are adjacent to v_0, so v_0 is the basis of the hypercube with signature $\mathcal{E}^-(u)$ and anti-basis u. A shortest (v_0, u)-path is thus made up of edges of this hypercube. The distance $d(v_0, u)$ is equal to its dimension: $d(v_0, u) = |\mathcal{E}^-(u)|$.

A key result is the fact that the central vertex v_0 of the simplex graph belongs to the interval $I(u, v)$ of any pair u, v satisfying $d(u, v) = \mathsf{ecc}(u)$.

▶ **Theorem B.4.** *Let $u, v \in V(G)$ such that $d(u, v) = ecc(u)$. Then, $v_0 \in I(u, v)$.*

Proof. Suppose, by way of contradiction, that $m = m(u, v, v_0) \neq v_0$. There is at least one edge e incoming into m according to the v_0-orientation. Let E_i denote its Θ-class: $E_i \in \sigma_{v_0,m}$. Let E_j be a Θ-class separating m and v: $E_j \in \sigma_{m,v}$. We have $i \neq j$: if $E_i \in \sigma_{m,v}$, then the concatenation of a shortest (v_0, m)-path with a shortest (m, v)-path does not give a shortest (v_0, v)-path (Lemma A.2), a contradiction as $m \in I(v_0, v)$. Let $v' \in \partial H'_j$ be some vertex adjacent to an edge of E_j in $I(m, v)$. According to Theorem B.2, E_j is also adjacent to v_0. In summary, $v_0, v' \in \partial H'_j$. By convexity of the boundaries (Lemma 2.5), all vertices of $I(v_0, v')$ belong to $\partial H'_j$, among them the two endpoints of e because $I(v_0, m) \subseteq I(v_0, v')$. As E_j consists in an isomorphism between its two boundaries and $e \in E_i$, then $E_i \perp E_j$.

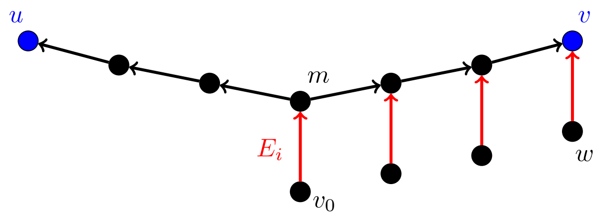


Figure 7 Illustration of the contradiction in the proof of Theorem B.4.

As a consequence, E_i is orthogonal to any Θ-class of $\sigma_{m,v}$. We remind that E_i is adjacent to m. By successive applications of Lemma A.5 to the vertices of a shortest (m, v)-path, we show that E_i is adjacent to all these vertices, in particular v. Here comes the contradiction: any shortest (u, v)-path can be extended by concatenating to it the edge of E_i adjacent to v, as $E_i \notin \sigma_{u,v}$. So, $v_0 = m$ and $v_0 \in I(u, v)$. ◀

Figure 7 shows a simple example with $d(v_0, m) = 1$. The edges are oriented according to the v_0-orientation. The Θ-class E_i in $\sigma_{v_0,m}$ is present alongside the interval $I(m, v)$. The contradiction of the previous proof comes from the fact that the shortest (u, v)-path could be extended with the vertex w which is the neighbor of v in $\partial H'_i$.

Two vertices u, v forming a diametral pair cannot share a common incoming Θ-class E_i, in other words $\mathcal{E}^-(u) \cap \mathcal{E}^-(v) = \emptyset$, otherwise $m = m(u, v, v_0) \in I(u, v) \subseteq H''_i$ and $v_0 \in H'_i$. Moreover, the distance $d(u, v)$ is exactly $|\mathcal{E}^-(u)| + |\mathcal{E}^-(v)|$ because $|\mathcal{E}^-(u)| = d(v_0, u)$ and $|\mathcal{E}^-(v)| = d(v_0, v)$. So, determining the diameter of a simplex graph G is equivalent to maximizing the sum $|X| + |Y|$, where X and Y are two POFs of G that are disjoint. Computing the diameter is equivalent to find the largest pair of disjoint cliques in the crossing graph $G^\#$. Similarly, the eccentricity of a vertex u is exactly the size $|\mathcal{E}^-(u)| + |\mathcal{E}^-(v)|$ of the largest pair of disjoint POFs $(\mathcal{E}^-(u), \mathcal{E}^-(v))$. Now, we can define the notion of *opposite*.

▶ **Definition B.5.** *Let G be a simplex graph and X a POF of G. We denote by $\mathsf{op}(X)$ the opposite of X,* i.e. *the POF Y disjoint from X with the maximum cardinality.*

$$\mathsf{op}(X) = \operatorname*{argmax}_{Y \cap X = \emptyset} |Y| \,.$$

With this definition, the eccentricity of a vertex u, if we fix $X_u = \mathcal{E}^-(u)$, is written $\mathsf{ecc}(u) = |X_u| + |\mathsf{op}(X_u)|$. Hence, the diameter of the simplex graph G can be written as the size of the largest pair POF-opposite: $\mathsf{diam}(G) = \max_{X \in \mathcal{L}}(|X| + |\mathsf{op}(X)|)$.

We propose now the definition of two problems on simplex graphs. The first one, called Opposites (OPP) consists in finding all pairs POF-opposite. Its output has thus a linear size. Given the solution of OPP on graph G, one can deduce both the diameter and all eccentricities in $O(n)$ time with the formulae presented above.

▶ **Definition B.6** (OPP).

Input: *Simplex graph G, central vertex v_0.*

Output: *For each POF X, its opposite $\mathsf{op}(X)$.*

We define an even larger version of the problem where a positive integer weight is associated with each POF. We call it Weighted Opposites (WOPP).

▶ **Definition B.7** (WOPP).

Input: *Simplex graph G, central vertex v_0, weight function $\omega : \mathcal{L} \to \mathbb{N}^+$.*

Output: *For each POF X, its weighted opposite Y maximizing $\omega(Y)$ such that $X \cap Y = \emptyset$.*

Obviously, OPP is a special case of WOPP when ω is the cardinality function. In Section B.2, we show that WOPP can be solved in quasilinear time $O((d^3 + \log n)n)$. As a consequence, all eccentricities of a simplex graph G can also be determined with such time complexity. Moreover, we will see in Section 4 that solving WOPP in quasilinear time implies that all eccentricities of any median graph can be computed with a simple exponential time $2^{O(d)}n$, improving the slightly super-exponential time proposed in [14].

B.2 Quasilinear algorithm for all eccentricities in simplex graphs

We propose an algorithm solving WOPP in quasilinear time $\tilde{O}(n)$.

▶ **Theorem B.8.** *There is a combinatorial algorithm solving WOPP in time $O((d^3 + \log n)n)$.*

Thus, we can compute the diameter and all eccentricities of simplex graphs in quasilinear time, even when the dimension is not bounded.

▶ **Corollary B.9.** *There is a combinatorial algorithm determining all eccentricities of a simplex graph in time $O((d^3 + \log n)n)$.*

The entire subsection is the proof of Theorem B.8. We consider a simplex graph G with a central vertex v_0 and a weight function $\omega : \mathcal{L} \to \mathbb{N}^+$. The algorithm is presented in Sections B.2.1 and B.2.2 and its analysis in Section B.2.3.

B.2.1 Tree structure of the opposites

The first step of our algorithm consists in building a binary tree T. Tree T is a representation of a partition refinement procedure over $\mathcal{L}$. We remind that partition refinement is a powerful algorithmic technique leading to the design of linear-time algorithms for many well-known problems [25, 38]. It consists in successive partitionings of a collection of sets.

In our context, the collection which is splitted is the set of POFs $\mathcal{L}$ of the simplex graph G. The vertices $a \in V(T)$ of tree T, called *nodes*, represent the sets obtained from the successive partitionings. They are indexed with POFs $L_a \in \mathcal{L}$. The edges of T are indexed with a pair class-boolean. For each $a \in V(T)$ with two children, the two edges connecting it to his children are both indexed with the same Θ-class but not the same boolean. This Θ-class is denoted by $E[a]$. To improve the readibility, index $(E[a], \text{true})$ will be denoted by $+E[a]$ while $(E[a], \text{false})$ becomes $-E[a]$.

We denote by $\mathcal{L}[E_i]$ the *adjacency list* of Θ-class E_i, *i.e.* the list of POFs in $\mathcal{L}$ which contain E_i. As at most d classes belong to any POF, the total size of all adjacency lists is upper-bounded by $dn = d|\mathcal{L}|$.

We sort the POFs in $\mathcal{L}$ in function of their weights $\omega(L)$ in the decreasing order. This takes $O(n \log n)$. We denote by τ_ω this ordering. Let L_0 be the maximum-weighted POF in $\mathcal{L}$, in other words the first POF in ordering τ_ω.

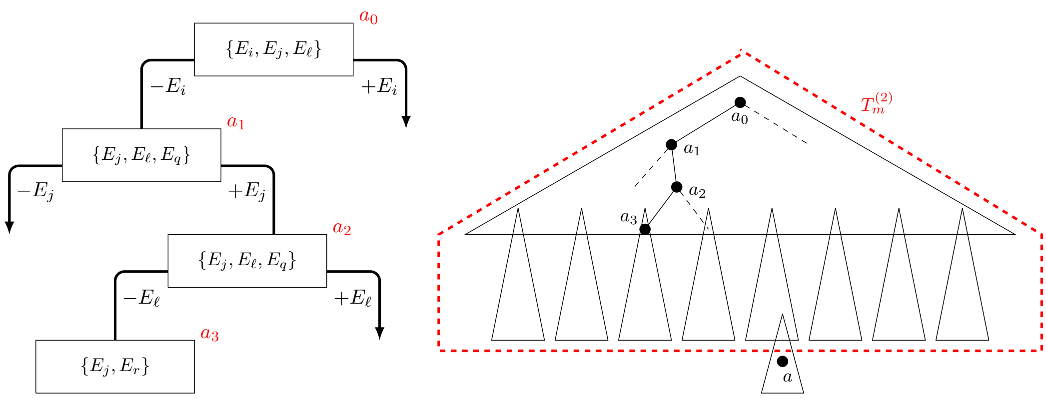
The description of the construction of T by partition refinement begins. We assign an arbitrary ordering to the Θ-classes which are in L_0, for example based on their index. Let $E_{i_1}, E_{i_2}, \ldots, E_{i_r}$ be the classes of L_0 ordered, $|L_0| = r$. First, we split $\mathcal{L}$ in two sets: one with POFs containing E_{i_i}, the other with POFs which do not contain E_{i_i}. In brief, we obtain $\mathcal{L}[E_{i_1}]$ and its complementary. Second, we split each of these two sets regarding Θ-class E_{i_2}: on one side the POFs containing E_{i_2}, on the other side POFs without E_{i_2}. We pursue in this way with all classes of L_0. At the end, there are at most $2^{|L_0|}$ sets in the partition.

Until now, we obtained the top r depths of tree T. Before pursuing the construction, we define some notation. Let a_0 be the root of T. Node a_0 represents the entire collection $\mathcal{L}$. Let Ω be the function indicating the sets represented by each node $a \in V(T)$. We have $\Omega(a_0) = \mathcal{L}$. For any $a \in T$, its index L_a is defined as the maximum-weighted POF of its *universe* $\Omega(a)$. So, $L_{a_0} = L_0$. The root a_0 has two children, one representing $\mathcal{L}[E_{i_1}]$ and the other one its complementary as they are the result of a partition refinement from E_{i_1}. The edges connecting a_0 and its children are indexed by $+E_{i_1}$ and $-E_{i_1}$ respectively.

We denote by $R(a)$ the set of edge indices of the simple path from a to the root a_0. For example, for node a with a universe $\Omega(a)$ being the set of POFs which contain E_{i_2} but not E_{i_1}, we have $R(a) = \{E_{i_1}, E_{i_2}\}$. We write $R(a) = R^+(a) \cup R^-(a)$, where $R^+(a)$ contains indices with boolean true, while $R^-(a)$ contains indices with boolean false. In this example, $R^+(a) = \{E_{i_2}\}$ and $R^-(a) = \{E_{i_1}\}$. Set $R^+(a)$ is a POF: it contains Θ-classes which are pairwise orthogonal, otherwise $R^+(a)$ would not be the subset of a POF belonging to $\mathcal{L}$.

We pursue the construction of T. For each leaf a of the current tree, we execute the following process: we only consider the Θ-classes of L_a which have not been treated earlier, *i.e.* which are not in $R^+(a)$. We order them arbitrarily and we split the universe $\Omega(a)$ successively. We pursue with the new leaves obtained, etc. In this way, tree T can be seen as a stacking of small binary trees (depth at most d) that we call *blocks*. For example, the root a_0 of T belongs to the top block which is produced from the partition refinement over Θ-classes of L_0. In Figure 8a, we represent some nodes a of this block and their index L_a with $|L_0| = 3$: $i_1 = i$, $i_2 = j$, and $i_3 = \ell$. The leaves of this block both belongs to the top block and are the roots of another block below this one.

Algorithm 1 provides us with the pseudocode of a partition refinement procedure. We



(a) Some nodes of T obtained with the partition refinement over L_0

(b) Some blocks of T: all blocks of layer 0 and 1, one block of layer 2. Node a has layer 2.

Figure 8 An example of tree T: its first block and structure

call it Ordered Internal Partition Refinement (OIPR). We execute OIPR to obtain tree T with the following inputs: the ground set contains the Θ-classes of G ($W = \mathcal{E}$), the collection is made up the POFs outgoing from m ($\mathcal{S} = \mathcal{L}$) and the ordering of the sets come from the POFs weights ($\tau = \tau_\omega$). Each while loop (line 4) corresponds to the construction of a block of T. Such step starts by picking up the first element, say L_a, of a non-singleton set of $\mathcal{P}$, say $\Omega(a)$. Then, we consider the Θ-classes of L_a which have not been locally visited, *i.e.* which do not belong to $R^+(a)$ (line 7). For each of these classes E_j, we refine the non-singleton $\Omega(a)$: we split it successively with the POFs containing E_j and the POFs not containing E_j (line 8).

Algorithm 1 Ordered Internal Partition Refinement (OIPR)

1: **Input:** ground set $W = \{w_1, \ldots, w_q\}$, collection $\mathcal{S} = \{S_1, \ldots, S_N\}$, $S_i \subseteq W$ for any $1 \leq i \leq N$, and an ordering τ of $\mathcal{S}$.
2: **Output:** An ordered partition $\mathcal{P}$ of $\mathcal{S}$ made up of singletons.
3: Initialize $\mathcal{P} \leftarrow \{\mathcal{S}\}$, partition with a single set;
4: **while** *there exists a part of* $\mathcal{P}$ *which is not a singleton* **do**
5: $\quad Q \leftarrow$ first non-singleton of the ordered partition $\mathcal{P}$;
6: $\quad A \leftarrow$ first element of Q according to τ;
7: $\quad$ **for** *every* $w_j \in A$ *non locally visited* **do**
8: $\quad\quad$ Substitute Q in $\mathcal{P}$ by $\mathsf{Refine}(Q, \{w_j\})$;
9: $\quad$ **endfor**
end

The time needed to run Algorithm 1, using a doubly linked list data structure, depends on the number of appearances of each element w_j of the ground set into the collection $\mathcal{S}$.

▶ **Lemma B.10** (Execution time of OIPR [25]). *Let $M(w_j)$ be the number of sets $S_i \in \mathcal{S}$ such that $w_j \in S_i$. Then, OIPR runs in $O(\sum_{j=1}^{q} M(w_j))$.*

In our context, values $M(w_j)$ are the sizes of adjacency lists $\mathcal{L}[E_j]$. We explained above why the total size of these adjacency lists could not exceed dn.

▶ **Corollary B.11.** *OIPR applied with $W = \mathcal{E}$, $\mathcal{S} = \mathcal{L}$ and $\tau = \tau_\omega$ runs in $O(n(d + \log n))$.*

Proof. The time needed to sort all POFs according to their weights, *i.e.* in order τ_ω, is $O(n \log n)$. Moreover, OIPR runs in $O(dn)$ according to Lemma B.10. ◀

Each set obtained from a refinement is represented by a node of T. Its children are obtained from a refinement of $\Omega(a)$ with some Θ-class $E_j = E[a]$: one represents the elements of $\Omega(a)$ containing E_j (the edge from a to this child is indexed with $+E_j$), the other represents the complementary (the edge from a to this child is indexed with $-E_j$).

Now, we give some notation and properties related to the tree T. At least one partitioning is executed at each depth of tree T so, as $\mathcal{L}$ is finite, T is too. Its depth is at most $n = |\mathcal{L}|$. We say the *layer* of a node a is the number of blocks we pass through when we traverse the simple path between the root and a minus 1. For example, the root has layer 0 and the leaves of the top block have layer 1. We denote by $T^{(j)}$, $j \geq 1$, the subtree of T induced on the set made up of (i) nodes of layer at most $j-1$ and (ii) nodes of layer j which are the roots of a block. In this way, nodes of layer j in $T^{(j)}$ are leaves. The depth of $T^{(j)}$ is at most jd, as the depth of each block is at most d.

Certain nodes $a \in V(T)$ may admit only one child. This situation occurs when $E[a]$ is not orthogonal to at least one class of $R^+(a)$. Indeed, each POF of $\Omega(a)$ contains all Θ-classes of $R^+(a)$: if $R^+(a) \cup \{E[a]\}$ is not a POF, then there is no POF superset of $R^+(a) \cup \{E[a]\}$.

For any node a, we store its universe $\Omega(a)$. With the doubly linked list data structure for each partitioning, we can preserve the original ordering of the POFs. Consequently, in all sets $\Omega(a)$, the POFs are sorted in function of their weight. For any $a \in V(T)$, the maximum-weighted POF of $\Omega(a)$, *i.e.* index L_a, is thus obtained with constant running time.

Even if the execution time of OIPR is quasilinear in n, the extra time needed to store the tree T and particularly all sets $\Omega(a)$ may not be linear in n. In the remainder, we will see that considering tree $T^{(d)}$ suffices to solving our problem. In this way, storing $T^{(d)}$ becomes quasilinear in n, such as the execution of the partitioning.

B.2.2 Constraint pairs

The second step of our algorithm uses a data structure called *constraint pair*, whose definition is based on tree T. We provide a dynamic programming (DP) algorithm which computes one value per constraint pair. At the end of the execution, we can deduce the opposite of each POF in $O(1)$.

▶ **Definition B.12** (Constraint pair). *A constraint pair (a, X) is made up of a node $a \in V(T)$ and a POF X such that* (i) $X \cap R(a) = \emptyset$ *and* (ii) $X \cup R^+(a) \in \mathcal{L}$.

The existence of a constraint pair (a, X) implies that no class X is present in the edge indices from a_0 to a in T. Moreover, each Θ-class of X is orthogonal to all Θ-classes of $R^+(a)$. We denote by $\mathcal{C}$ the set of constraint pairs. Let $\mathcal{C}^{(j)}$ be the set of constraint pairs (a, X) such that $a \in T^{(j)}$: we have $\mathcal{C}^{(j)} \subseteq \mathcal{C}^{(j+1)} \subseteq \mathcal{C}$.

For any constraint pair (a, X), we denote by $h(a, X)$ a POF X^* disjoint from X in $\Omega(a)$ with the maximum weight. For any POF $L \in \mathcal{L}$, the POF $h(a_0, L)$ is an opposite of L because $\Omega(a_0) = \mathcal{L}$, so we can write, according to Definition B.5, $h(a_0, L) = \mathsf{op}(L)$. Observe that any pair (a_0, L) is a constraint pair as $R(a) = \emptyset$. We present a method to compute all opposites $h(a_0, L)$.

The description of the DP algorithm starts. For any constraint pair (a, X), we denote by a^+ (resp. a^-) the child of a which is the endpoint of the edge indexed with $+E[a]$ (resp. $-E[a]$) in T (Figure 9). We define the set $\mathcal{C}(a, X)$ which describes the recursive calls needed to compute $h(a, X)$. Formally, the objective is to make sure that value $h(a, X)$ is exactly a function of all $h(a', X')$, where $(a', X') \in \mathcal{C}(a, X)$. The construction of set $\mathcal{C}(a, X)$ is described below. We distinguish four cases: A, B, C, and D.

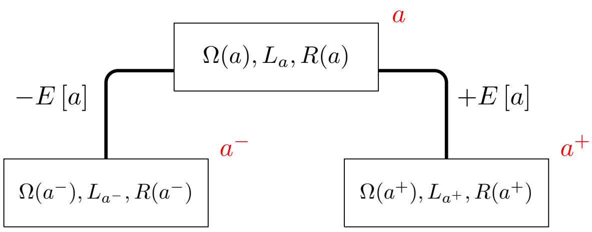


Figure 9 Nodes a, a^+ and a^- in tree T.

- **Case A**. No class of X is in L_a, *i.e.* $L_a \cap X = \emptyset$. Otherwise, see next cases.
  As L_a is the maximum-weighted POF in $\Omega(a)$ and $L_a \cap X = \emptyset$, we have $h(a, X) = L_a$. No recursive call is needed: $\mathcal{C}(a, X) = \emptyset$.
  A special case of Case A is when $\Omega(a)$ is a singleton: $\Omega(a) = R^+(a)$. We know from Definition B.12 that $R^+(a) \cap X = \emptyset$.
- **Case B**. Class $E\,[a]$ belongs to X: $E\,[a] \in X$. Otherwise, see next cases.
  As $R(a^-) = R(a) \cup \{E\,[a]\}$, $(a^-, X \backslash (E\,[a]))$ is a constraint pair (Definition B.12): $R(a^-) \cap (X \backslash (E\,[a])) = \emptyset$ and $R^+(a^-) = R^+(a)$ is orthogonal to all Θ-classes of set $X \backslash (E\,[a]) \subsetneq X$. We fix $\mathcal{C}(a, X)$ as a singleton containing $(a^-, X \backslash (E\,[a]))$: $\mathcal{C}(a, X) = \{(a^-, X \backslash (E\,[a]))\}$.
- **Case C**. Set $E\,[a] \cup X$ is a POF: $E\,[a] \cup X \in \mathcal{L}$.
  As $E\,[a] \notin X$, we have $X \cap R(a^+) = X \cap R(a^-) = \emptyset$. First, $R^+(a^-) = R^+(a)$, so $X \cup R^+(a^-)$ is a POF and $(a^-, X) \in \mathcal{C}$. Second, $R^+(a^+) = R^+(a) \cup E\,[a]$: as $E\,[a]$ is orthogonal to all Θ-classes of X, $R^+(a^+) \cup X$ is a POF and $(a^+, X) \in \mathcal{C}$.
  These two constraint pairs are the elements of $\mathcal{C}(a, X)$: $\mathcal{C}(a, X) = \{(a^+, X), (a^-, X)\}$.
- **Case D**. Set $E\,[a] \cup X$ is not a POF: $E\,[a] \cup X \notin \mathcal{L}$.
  Let $X_{|E[a]} \subsetneq X$ be the Θ-classes of X which are orthogonal to $E\,[a]$. Pair (a^-, X) is a constraint pair, using the same arguments than in Case C. We verify whether another pair $(a^+, X_{|E[a]}) \in \mathcal{C}$. As $X_{|E[a]} \subsetneq X$ and $E\,[a] \notin X$, we have $R(a^+) \cap X_{|E[a]} = \emptyset$. Furthermore, $X_{|E[a]} \cup R^+(a)$ is a POF because (a, X) is a constraint pair and $X_{|E[a]} \cup E\,[a]$ is a POF by definition, so $X_{|E[a]} \cup R^+(a^+)$ is a POF.
  We fix $\mathcal{C}(a, X) = \left\{(a^-, X), (a^+, X_{|E[a]})\right\}$.

Observe that when $(a', X') \in \mathcal{C}(a, X)$, then a is a parent of a' in T. Moreover, $X' \subseteq X$. The size of all sets of recursive calls $\mathcal{C}(a, X)$ is at most two. The following theorem justifies that POF $h(a, X)$ can be determined as a function of all $h(a', X')$ satisfying $(a', X') \in \mathcal{C}(a, X)$.

▶ **Theorem B.13.** *Let $(a, X) \in \mathcal{C}$. If $\mathcal{C}(a, X) = \emptyset$ (Case A), $h(a, X)$ is equal to L_a. Otherwise:*

$$h(a, X) = \underset{h(a', X') \ \ s.t. \ (a', X') \in \mathcal{C}(a, X)}{\operatorname{argmax}} \omega\left(h(a', X')\right) \tag{3}$$

Proof. The justification for Case A was evoked above: as $L_a \cap X = \emptyset$ and L_a is the maximum-weighted POF of $\Omega(a)$, then $h(a, X) = L_a$.

In Case B, $E\,[a] \in X$. We remind that $\mathcal{C}(a, X) = \{(a^-, X \backslash E\,[a])\}$ in this case. POF $h(a, X)$, which is disjoint from X, cannot contain class $E\,[a]$. As a consequence, $h(a, X)$

belongs to $\Omega(a^-)$ which is made up of the POFs of $\Omega(a)$ without $E[a]$. Moreover, $h(a,X)$ does not contain any class of $X \backslash E[a]$. We have: $h(a,X) = h(a^-, X \backslash E[a])$.

In Case C, we assume that $E[a] \cup X$ is a POF. As $E[a] \notin X$, $h(a,X)$ can be either in $\Omega(a^-)$ or in $\Omega(a^+)$. POFs $h(a^-,X)$ and $h(a^+,X)$ are respectively the maximum-weighted POFs of $\Omega(a^-)$ and $\Omega(a^+)$ without any class of X. Equation (3) holds as $\mathcal{C}(a,X) = \{(a^-,X),(a^+,X)\}$.

In Case D, we assume that $E[a] \cup X$ is not a POF. As in Case C, $E[a] \notin X$, so (a^-,X) is a constraint pair. If $h(a,X)$ belongs to $\Omega(a^-)$, then it is $h(a^-,X)$. Moreover, $(a^+, X_{|E[a]}) \in \mathcal{C}$ and we prove that if $h(a,X)$ belongs to $\Omega(a^+)$, it is $h(a^+, X_{|E[a]})$. Let E_h be a Θ-class of $X \backslash X_{|E[a]}$. Classes E_h and $E[a]$ are parallel, otherwise E_h would belong to $X_{|E[a]}$. All POFs of set $\Omega(a^+)$ contain $E[a]$, so none of them can contain E_h: it would be contradictory with the non-orthogonality of these Θ-classes. In summary, no POF in $\Omega(a^+)$ contains a Θ-class of $X \backslash X_{|E[a]}$. Therefore, determining the maximum-weighted POF of $\Omega(a^+)$ disjoint from X is equivalent to finding the maximum-weighted POF of $\Omega(a^+)$ disjoint from $X_{|E[a]}$. In brief, if $h(a,X)$ is in $\Omega(a^+)$, it is $h(a^+, X_{|E[a]})$. Equation (3) holds as $\mathcal{C}(a,X) = \left\{(a^-,X),(a^+,X_{|E[a]})\right\}$. ◀

The DP algorithm consists in recursively applying Equation (3) from all $h(a_0,L)$, for any $L \in \mathcal{L}$ and store the POFs $h(a,X)$ which are computed throughout the execution. Case A is the base case of the recursion. Let H be the directed acyclic graph (DAG) representing the recursive calls of our DP. Its vertex set $V(H)$ contains all $(a,X) \in \mathcal{C}$ such that $h(a,X)$ is called for the computation of certain POFs $h(a_0,L)$. Its edge set is made up of arcs from $(a,X) \in V(H)$ to elements in $\mathcal{C}(a,X)$. Certain constraint pairs of $\mathcal{C}$ may not belong to $V(H)$, *i.e.* they are not needed to compute the opposites. Figure 10 illustrates the DAG H with a vertex (a,X) and its nearby successors. In this example, pair (a,X) is needed for the computation of $\mathsf{op}(L) = h(a_0,L)$ and $\mathsf{op}(\widehat{L}) = h(a_0,\widehat{L})$: dotted lines mean there is a path between two pairs.

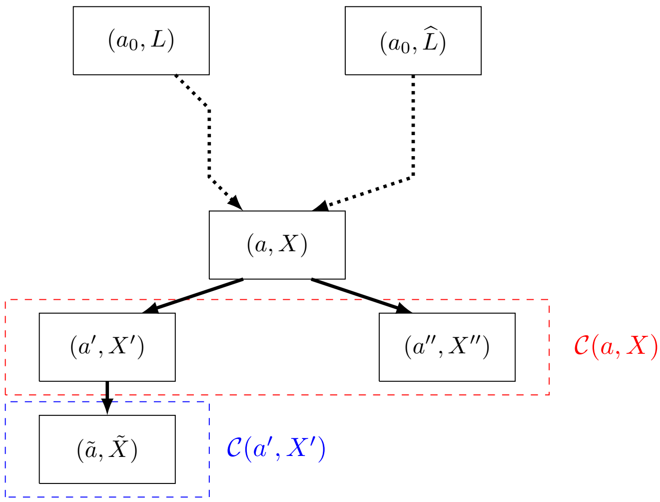


Figure 10 Some vertices of the DAG H

We observe that the constraint pairs of $V(H)$ are made up of nodes of $T^{(d)}$.

▶ **Lemma B.14.** *Constraint pairs (a,X) in $V(H)$ satisfy the following inequality: $|X| \leq d - \mathsf{layer}(a)$.*

Proof. We proceed by induction. The roots of the DAG, *i.e.* the constraint pairs (a_0,X), verify this inequality, as $\mathsf{layer}(a_0) = 0$.

Assume now that the ancestors of $(a, X) \in V(H)$ satisfy the inequality. We distinguish two cases. First, suppose there is a predecessor of (a, X), *i.e.* a constraint pair (a', X') and an arc going from it to (a, X), such that $\mathsf{layer}(a) = \mathsf{layer}(a') + 1$. We select a pair $(\widehat{a}', \widehat{X}')$ in $V(H)$ which is an ancestor of (a', X') - there is a directed path from $(\widehat{a}', \widehat{X}')$ to (a', X') in H - and such that $\widehat{a}'$ is the ancestor of a' in T which is at the top of the block $\mathcal{B}_{\widehat{a}'}$ of layer $\mathsf{layer}(a')$ containing a'. Thus, node a is a leaf of block $\mathcal{B}_{\widehat{a}'}$. The existence of a directed path in H from $(\widehat{a}', \widehat{X}')$ to (a, X) means that the recursive computation of $h(\widehat{a}', \widehat{X}')$ uses the term $h(a, X)$. The POF $L_{\widehat{a}'}$ is not disjoint from $\widehat{X}'$ otherwise pair $(\widehat{a}', \widehat{X}')$ would be a leaf in H (Case A). Let $E_h \in L_{\widehat{a}'} \cap \widehat{X}'$. As a is a leaf of the block $\mathcal{B}_{\widehat{a}'}$, it verifies $E_h \in R(a)$ by construction of T. As $(\widehat{a}', \widehat{X}')$ is an ancestor of (a, X), then $X \subseteq \widehat{X}'$. But (a, X) is a constraint pair and $E_h \in R(a)$, so $X \subseteq \widehat{X}' \setminus \{E_h\}$. In brief, $|X| \leq \left|\widehat{X}'\right| - 1$. Using the induction hypothesis, $|X| \leq d - \mathsf{layer}(\widehat{a}') - 1 = d - \mathsf{layer}(a)$.

Second, suppose that all predecessors (a', X') of (a, X) satisfy $\mathsf{layer}(a') = \mathsf{layer}(a)$. As $X \subseteq X'$, $|X| \leq |X'| \leq d - \mathsf{layer}(a') = d - \mathsf{layer}(a)$. ◀

As a consequence, for any pair $(a, X) \in V(H)$, the depth of node a in T can be upper-bounded by d^2 because each block has at most depth d and the layer of a is at most d. Formally, $V(H) \subseteq \mathcal{C}^{(d)}$. This shows that computing tree $T^{(d)}$ - and getting rid of the larger depths of the tree - is sufficient for the execution of the DP. This observation allows us to state in the next subsubsection that this algorithm runs in quasilinear time.

B.2.3 Analysis

In this section, we prove that the DP algorithm described earlier can run in quasilinear time $O(d^3 n)$. The global algorithm is divided into two parts.

1. Construction of the tree $T^{(d)}$ with maximum layer d, using partition refinement, storage of $\Omega(a)$, L_a, and $R(a)$ for each node a of $T^{(d)}$.
2. Computation of all opposites $h(a_0, L)$ with DP.

The runtime needed to build $T^{(d)}$ is $O(d^2 n)$. The execution time of all partitionings is $O((d + \log n)n)$, according to Corollary B.11. The memory space used to store sets $\Omega(a), L_a, R(a)$ for each node $a \in V(T^{(d)})$ can be upper-bounded by $O(d^2 n)$ because the sets $\Omega(a)$ together at some fixed depth form a partition of $\mathcal{L}$ and the depth of $T^{(d)}$ is at most d^2.

The analysis for the second part of the algorithm is not trivial. Our key argument consists in providing an upper bound for the number of constraint pairs in $\mathcal{C}^{(d)}$. It provides us with a maximum number of POFs which must be stored during the DP.

▶ **Theorem B.15.** *There are at most $d^2 n$ constraint pairs in $\mathcal{C}^{(d)}$.*

Proof. We define a function $f : \mathcal{L} \to 2^{\mathcal{C}}$ such that, for any $L \in \mathcal{L}$, $|f(L)| \leq d^2$. We show that for any constraint pair $(a, X) \in \mathcal{C}^{(d)}$, there is a POF Y, $X \subseteq Y$, such that $(a, X) \in f(Y)$. As the total number of constraint pairs generated by f is at most $d^2 n$, we have $\left|\mathcal{C}^{(d)}\right| \leq d^2 n$.

- Step 1: definition of function f. Our method consists in defining an *itinerary function* $I : \mathcal{C}^{(d)} \to \mathcal{C}^{(d)}$, where $(a, X) = I(a', X')$ implies that $X \subseteq X'$ and a is a child of a' in $T^{(d)}$. Starting from some (a_0, Y), where $Y \in \mathcal{L}$, successive appliances of function I provide us with a descent in $T^{(d)}$, *i.e.* a simple path from a_0 to a leaf of $T^{(d)}$, if we refer to the nodes of the constraint pairs obtained. Function f will be defined as follows: $f(Y) = \{(a_0, Y), I(a_0, Y), I(I(a_0, Y)), \ldots\}$. As the depth of $T^{(d)}$ is at most d^2, we confirm that $|f(Y)| \leq d^2$.

Here is the definition of function I and we verify the properties announced above. Let $(a', X') \in \mathcal{C}^{(d)}$: we denote by a^- (resp. a^+) his child which is the endpoint of edge indexed by $-E[a']$ (resp. $+E[a']$). We distinguish three cases: as they are similar to some Cases enumerated the previous subsubsection, we denote them by B*, C* and D* respectively.

- **Case B***: Θ-class $E[a']$ belongs to X, *i.e.* $E[a'] \in X$. We fix $I(a', X') = (a^-, X \setminus \{E[a']\})$.
- **Case C***: $E[a'] \notin X$ and $E[a'] \cup X$ is a POF. We fix $I(a', X') = (a^+, , X)$.
- **Case D***: $E[a'] \notin X$ and $E[a'] \cup X$ is not a POF. We fix $I(a', X') = (a^-, X)$.

Any constraint pair is concerned by one of these three cases. One can see that if $I(a', X') = (a, X)$ then a is a child of a' and $X \subseteq X'$.

- Step 2: any constraint pair is generated by f. We show that any $(a, X) \in \mathcal{C}^{(d)}$ can be written as $I^k(a_0, Y)$ for some POF $Y \supseteq X$ and $0 \le k < d^2$. We proceed by induction on the depth of a. The base case is trivial: If $\mathsf{depth}(a) = 0$, then $a = a_0$. For any $X \in \mathcal{L}$, $(a_0, X) \in f(X)$.

Let $(a, X) \in \mathcal{C}^{(d)}$, $a \neq a_0$, we assume that each constraint pair containing an ancestor of a belongs to some $f(Y)$. We distinguish three scenarii depending on the nature of both the parent a' of a and Θ-class $E[a']$. For each scenario, we show that there is a constraint pair (a', X') such that $I(a', X') = (a, X)$ and $X \subseteq X'$. As $(a', X') = I^k(a_0, Y)$ and $X' \subseteq Y$ by induction, such statement implies that $(a, X) = I^{k+1}(a_0, Y)$ and $X \subseteq Y$, in other words $(a, X) \in f(Y)$.

- **Scenario 1**: edge (a', a) in T is indexed by $-E[a']$ and $X \cup E[a']$ is a POF.
  Let $X' = X \cup \{E[a']\}$. We begin with the proof that (a', X') is a constraint pair. First, set $R(a)$ is the union of singleton $\{E[a']\}$ with $R(a')$, so $E[a'] \notin R(a')$. As $(a, X) \in \mathcal{C}$, $X \cap R(a) = \emptyset$, so $X \cup \{E[a']\}$ has no intersection with $R(a')$. In brief, $X' \cap R(a') = \emptyset$. Second, as $(a, X) \in \mathcal{C}$, we affirm that any Θ-class of X is orthogonal to any class of $R^+(a) = R^+(a')$. Showing that $E[a']$ is orthogonal to any class of $R^+(a')$ will imply that $X' \cup R^+(a')$ is a POF. If a' is the root of some block, then $E[a']$ is a Θ-class of $L_{a'}$. As $L_{a'}$ contains all Θ-classes of $R^+(a')$ by definition, this shows the orthogonality of $E[a']$ with $R^+(a')$. Now, assume that a' is not the root of some block: $\mathcal{B}_{a'}$ denotes the block it belongs to and the root of $\mathcal{B}_{a'}$ is written $a^* \neq a'$ (Figure 11). Node a^* is an ancestor of a'. Set $R^+(a')$ can be splitted as follows: on one hand, the Θ-classes of $R^+(a')$ indexed between a_0 and a^*, *i.e.* $R^+(a^*)$, and on the other hand the Θ-classes of $R^+(a')$ whose indices appear in the block $\mathcal{B}_{a'}$, *i.e.* $R^+(a') \cap L_{a^*}$. Class $E[a']$ belongs to L_{a^*}, otherwise a' would be the leaf of this block and, therefore, the root of another block. Both $R^+(a^*)$ and $R^+(a') \cap L_{a^*}$ are subsets of L_{a^*}, so $R^+(a') \subsetneq L_{a^*}$. As a conclusion, $R^+(a') \cup \{E[a']\} \subseteq L_{a^*}$ is a POF, therefore $X' \cup R^+(a')$ is a POF and (a', X') is thus a constraint pair.
  We show that $I(a', X') = (a, X)$. We refer to Case B*: the Θ-class $E[a']$ belongs to X', so function I returns $(a^-, X' \setminus \{E[a']\}) = (a, X)$.
- **Scenario 2**: edge (a', a) in T is indexed by $-E[a']$ and $X \cup E[a']$ is not a POF.
  Let $X' = X$. As $(a, X) \in \mathcal{C}$ and $R(a') \subsetneq R(a)$, we have $X \cap R(a') = \emptyset$. Moreover, $R^+(a') = R^+(a)$, so $X \cap R^+(a')$ is a POF. In brief, $(a', X') = (a', X) \in \mathcal{C}$.
  We prove that $I(a', X') = (a, X)$. In this scenario, $X \cup \{E[a']\}$ is not a POF. Furthermore, $X \cap R(a) = \emptyset$ because $(a, X) \in \mathcal{C}$, so $E[a'] \notin X$. We refer to Case D* and $I(a', X') = (a^-, X') = (a, X)$.
- **Scenario 3**: edge (a', a) in T is indexed by $+E[a']$.
  Let $X' = X$. As $(a, X) \in \mathcal{C}$ and $R(a') \subsetneq R(a)$, we have $X \cap R(a') = \emptyset$ as in Scenario 2. Moreover, $R^+(a') \subsetneq R^+(a)$: only $E[a']$ is in $R^+(a)$ and not in $R^+(a')$. As $X \cup R^+(a)$ is a POF, its subset $X \cup R^+(a')$ is too. In brief, $(a', X') = (a', X) \in \mathcal{C}$.
  We show that $I(a', X') = (a, X)$. Class $E[a']$ is not in $X' = X$ because $X \cap R(a) = \emptyset$. Set $X \cup \{E[a']\}$ is a subset of the POF $X \cup R^+(a)$ because $E[a'] \in R^+(a)$, so $X \cup \{E[a']\}$

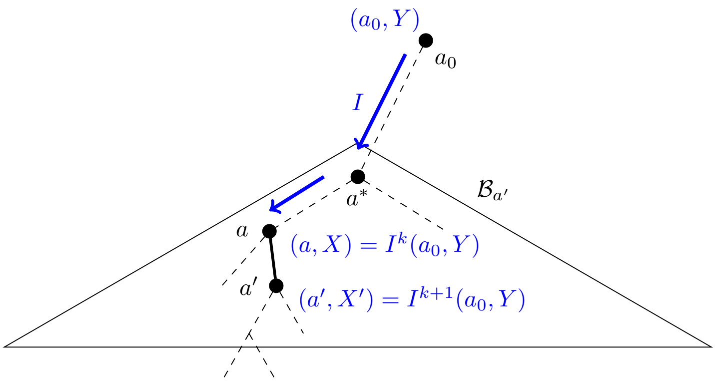


Figure 11 Nodes a, a', and a^* in T when a' is not the root of a block in Scenario 1. Blue arrows indicate the successive appliances of the itinerary function on (a_0, Y).

is a POF. We refer to Case C*: $I(a', X') = (a^+, X') = (a, X)$.

In summary, there is a constraint pair (a', X') such that a' is the parent of a, $X \subseteq X'$, and $I(a', X') = (a, X)$. The induction hypothesis terminates the proof of our claim: as (a', X') is in some itinerary $f(Y)$ with $X' \subseteq Y$, and $I(a', X') = (a, X)$, then (a, X) also belongs to $f(Y)$. As the size of each $f(Y)$, $Y \in \mathcal{L}$ is upper-bounded by d^2, the total number of constraint pairs is at most $d^2 n$. ◀

The size of the state space of the DP procedure is at most $O(d^3 n)$ because $|h(a, X)| \leq d$ for any $(a, X) \in V(H)$. For any pair $(a, X) \in V(H)$, at most two recursive calls are launched to compute $h(a, X)$. In brief, Theorem B.15 allows us to affirm that the execution time of the DP procedure is $O(d^3 n)$ and is thus the total running time of our algorithm.

C Appendix for Section 4

C.1 Linear FPT algorithm for constant-dimension median graphs

We remind in this subsection the different steps needed to obtain a linear-time algorithm computing all eccentricities of a median graph with constant dimension, $d = O(1)$. We show how the algorithm of Section B.2 can be integrated to it in order to improve the dependence on d. Let us begin with a reminder of the former result.

▶ **Lemma C.1** ([14]). *There is a combinatorial algorithm computing all eccentricities in a median graph G with running time $\tilde{O}(2^{d(\log d+1)} n)$.*

Some parts of this subsection are redundant with [14], however we keep this subsection self-contained. The new outcomes presented are Theorems C.10 and 4.1. The results that are reminded will also be useful for Section 5.

The algorithm evoked by Lemma C.1 consists in the computation of three kinds of labels: *ladder* labels φ, *opposite* labels op and *anti-ladder* labels ψ. The order in which they are given correspond to their respective dependence: op-labelings are functions of labels φ and ψ-labelings are functions of both labels φ and op. The definition of op-labelings on general median graphs is very close to the notion of opposite previously defined for simplex graphs in Section B.

C.1.1 Ladder labels

Some preliminary work has to be done before giving the definition of labels φ. We introduce the notion of *ladder set*. It is defined only for pairs of vertices u, v satisfying the condition $u \in I(v_0, v)$. In this situation, the edges of shortest (u, v)-paths are all oriented towards v with the v_0-orientation.

▶ **Definition C.2** (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 Θ-classes admitting an edge adjacent to u.*

Figure 12 shows a small median graph with four vertices v_0, u, v, x such that $u \in I(v_0, v)$ and $u \in I(v_0, x)$. It gives the composition of ladder sets $L_{u,v}$ and $L_{u,x}$.

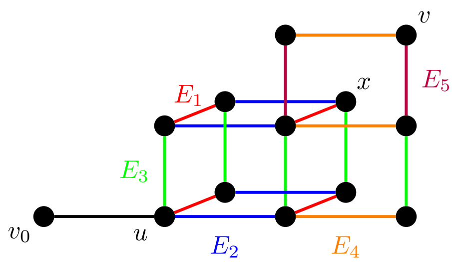


Figure 12 Examples of ladder sets: $L_{u,v} = \{E_2, E_3\}$, $L_{u,x} = \{E_1, E_2, E_3\}$.

A key characterization on ladder sets states that their Θ-classes are pairwise orthogonal. In brief, every set $L_{u,v}$ is a POF. Let us remind that the adjacency of all Θ-classes of a POF L with the same vertex u implies the existence of a (unique) hypercube not only signed with this POF L but also containing u (Lemma A.7). If additionnally POF L is *outgoing from* u - said differently, the edges adjacent to u belonging to a Θ-class of L leave u -, then u is the basis of the hypercube. As the Θ-classes of $L_{u,v}$ are adjacent to u by definition, there is a natural bijection between (i) hypercubes (ii) pairs made up of a vertex u and a POF L outgoing from u and (iii) pairs vertex-ladder set $(u, L_{u,\cdot})$.

▶ **Lemma C.3** ([14]). *Every ladder set $L_{u,v}$ is a POF. For any ordering τ of the Θ-classes in $L_{u,v}$, there is a shortest (u, v)-path such that, for any $1 \leq i \leq |L_{u,v}|$, the i^{th} first edge of the path belongs to the i^{th} Θ-class of $L_{u,v}$ in ordering τ.*

The necessary background to introduce labels φ is now known.

▶ **Definition C.4** (Labels φ [14]). *Given a vertex u and a POF L outgoing from u, let $\varphi(u, L)$ be the maximum distance $d(u, v)$ such that $u \in I(v_0, v)$ and $L_{u,v} = L$.*

Intuitively, integer $\varphi(u, L)$ provides us with the maximum length of a shortest path starting from u into "direction" L. Observe that the total size of labels φ on a median graph G does not exceed $O(2^d n)$, according to Lemma A.9. We provide another notion related to orthogonality which will be used in the remainder.

▶ **Definition C.5** (L-parallelism). *We say that a POF L' is* L-parallel *if, for any $E_j \in L'$, $L \cup \{E_j\}$ is not a POF.*

When L' is a L-parallel POF, we have $L \cap L' = \emptyset$, otherwise $L \cup \{E_j\} = L$ for some $E_j \in L'$. Presented differently, a L-paralle POF is such that any of its Θ-classes is parallel to at least one Θ-class of L.

A combinatorial algorithm running in $\tilde{O}(2^{2d}n)$ which computes all labels $\varphi(u, L)$ was identified in [14]: we provide an overview of it. There is a crucial relationship between a label $\varphi(u, L)$ and the labels of (i) the anti-basis u^+ of the hypercube with basis u and signature L and (ii) the L-parallel POFs outgoing from u^+.

▶ **Lemma C.6** (Inductive formula for labels φ [14]). *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) ladder set $L_{u^+,v}$ is L-parallel.*

A consequence of the previous lemma is that we can distinguish two cases for the computation of $\varphi(u, L)$. In the first case, $\varphi(u, L) = |L|$: it occurs when the farthest-to-u vertex with ladder set L is u^+ (base case). Indeed, u^+ is a candidate as $\sigma_{u,u^+} = L$: shortest (u, u^+)-paths pass through hypercube Q. This situation happens when either no Θ-class is outgoing from u^+ or when all Θ-classes outgoing from u^+ are orthogonal to L. In the second case, there are vertices farther to u than u^+ with ladder set L. As announced in Lemma C.6, $\varphi(u, L)$ is a function of labels $\varphi(u^+, \cdot)$.

$$\varphi(u, L) = \max_{\substack{L^+ \text{ POF outgoing from } u^+ \\ \forall E_j \in L^+,\ L \cup \{E_j\} \text{ not POF}}} (|L| + \varphi(u^+, L^+)). \tag{4}$$

If there exists such a POF L^+, then the label $\varphi(u, L)$ is given by Equation (4). Otherwise, it is given by the first case. Briefly, the algorithm consists in listing all pairs vertex-ladder set $((u, L), (u^+, L^+))$ such that u^+ is the anti-basis of the hypercube of basis u and signature L. For each of it, we verify whether L^+ is L-parallel. If it is, we update $\varphi(u, L)$ if $|L| + \varphi(u^+, L^+)$ is greater than the current value. The total number of pairs $((u, L), (u^+, L^+))$ is upper-bounded by $2^{2d}n$: there are at most $2^d n$ pairs (u^+, L^+) (bijection with hypercubes) and, for each of them, there are at most 2^d compatible pair (u, L) such that u^+ is the anti-basis of (u, L). Indeed, the number of edges ingoing into u^+ is at most d (Lemma A.8). For this reason, the computation of φ-labelings takes $\tilde{O}(2^{2d}n)$.

▶ **Theorem C.7** (Computation of labels φ [14]). *There is a combinatorial algorithm which determines all labels $\varphi(u, L)$ in $\tilde{O}(2^{2d}n)$. It also stores, for each pair (u, L), a vertex v satisfying $L_{u,v} = L$ and $d(u, v) = \varphi(u, L)$, denoted by $\mu(u, L)$.*

C.1.2 Opposite labels

The second type of labels needed to compute all eccentricities of a median graph G are opposite labels. Their definition is very close to the function op defined in Section B for simplex graphs. Given a vertex u and a POF L outgoing from u, let $\mathsf{op}_u(L)$ denote a POF with maximum label φ which is disjoint from L. As for φ, the total size of op-labelings is $O(2^d n)$.

▶ **Definition C.8** (Labels op [14]). *Let $u \in V$ and L be a POF outgoing from u. Let $\mathsf{op}_u(L)$ be one of the POF L' outgoing from u, disjoint from L, which maximizes value $\varphi(u, L')$.*

On simplex graphs, the opposite function provides in fact the op-labelings of vertex v_0: $\mathsf{op}(X) = \mathsf{op}_{v_0}(X)$. As all vertices belong to hypercubes with basis v_0, the ladder set $L_{v_0,v}$ for any vertex $v \in V$ is exactly the set $\mathcal{E}^-(v)$ of Θ-classes incoming into v. So, value $\varphi(v_0, X)$ is the distance $d(v_0, v)$ between v_0 and the only vertex v with ladder set $L_{v_0,v} = X$.

On general median graphs, the opposite label $\mathsf{op}_u(L)$ allows us to obtain the maximum distance $d(s, t)$ such that $u = m(s, t, v_0)$ and the ladder set $L_{u,s}$ is L.

▶ **Lemma C.9** (Relationship between medians and disjoint outgoing POFs [14]). *Let L, L' be two POFs outgoing from a vertex u. Let s (resp. t) be a vertex such that $u \in I(v_0, s)$ (resp. $u \in I(v_0, t)$) and $L_{u,s} = L$ (resp. $L_{u,t} = L'$). Then, $u \in I(s,t)$ if and only if $L \cap L' = \emptyset$. Therefore, given a single vertex s such that $u \in I(v_0, s)$ and $L_{u,s} = L$, the maximum distance $d(s, v)$ we can have with median $m(s, v, v_0) = u$ is exactly $d(u, s) + \varphi(u, \mathsf{op}_u(L))$.*

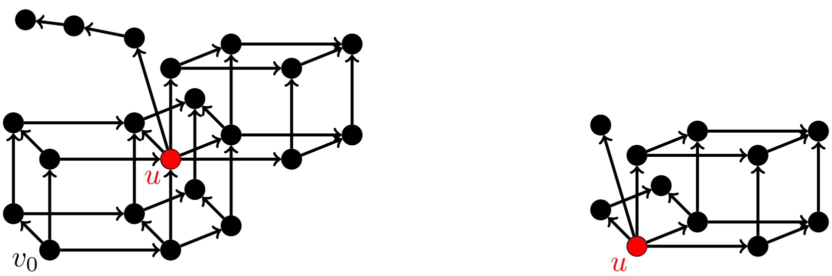
Going further, given a vertex $u \in V$, the maximum distance $d(s,t)$ such that $u = m(s, t, v_0)$ is the maximum value $\varphi(u, L) + \varphi(u, \mathsf{op}_u(L))$, where L is POF outgoing from u.

An algorithm had been initially proposed to compute all labels $\mathsf{op}_u(L)$ consisting in a brute force bounded tree search [14]. Its execution time was $\tilde{O}(2^{O(d \log d)} n)$, leading to the global same asymptotic running time (Lemma C.1) for finding all eccentricities.

Fortunately, the quasilinear time algorithm solving WOPP (Theorem B.8, Section B.2) offers us the opportunity to decrease the exponential term to a simple exponential function 2^d.

▶ **Theorem C.10** (Computation of labels op). *There is a combinatorial algorithm which determines all labels $\mathsf{op}_u(L)$ in $\tilde{O}(2^d n)$.*

Proof. Let $u \in V$: we denote by N_u the number of hypercubes of G with basis u. Let $G_u = G[V_u]$ be the *star* graph of u, using a definition from [21]. Its vertex set V_u is made up of the vertices belonging to an hypercube with basis u in G. Graph G_u is the induced subgraph of G on vertex set V_u (see Figure 13 for an example). Chepoi *et al.* [21] showed that graph G_u is a gated/convex subgraph of G. Naturally, convex subgraphs of median graphs are also median by considering the original definition of median graphs (Definition 2.1). In brief, G_u is median and all its maximal hypercubes contain a common vertex u. From Theorem B.2, G_u is a simplex graph.



(a) A v_0-oriented median graph G and a vertex $u \in V$ **(b)** Star graph G_u

Figure 13 Example of star graph G_u

Any pair (u, L) of G, where L is a POF outgoing from u in G, can be associated to a unique hypercube with signature L and basis u. Thus, there is a natural bijection between (i) the POFs of G_u (ii) the vertices of G_u and (iii) the POFs L of G outgoing from u. Hence, $|V_u| = N_u$.

We associate with any POF L of G_u the weight $\omega_u(L) = \varphi(u, L)$. We solve WOPP on graph G_u with weight function ω_u, using the algorithm evoked in Theorem B.8. The opposite computed with that configuration correspond exactly to the labels $\mathsf{op}_u(L)$: a POF L' disjoint from L and maximizing $\varphi(u, L')$ among all POFs outgoing from u. The running time of the algorithm is $O((d^3 + \log|V_u|)\,|V_u|) = O((d^3 + \log n)N_u)$. Doing it for every vertex u of G, we obtain all opposite labels of G in $O(d^3 + \log n)2^d n)$ as $\sum_{u \in V} N_u = 2^d n$ (Lemma A.9). ◀

C.1.3 Anti-ladder labels

We terminate with anti-ladder labels ψ which play the converse role of ladder labels φ. While $\varphi(u, L)$ is defined for POFs L outgoing from u, labels $\psi(u, R)$ are defined for POFs R ingoing into u, *i.e.* every Θ-class of the POF has an edge entering u. As any such pair (u, R) can be associated with an hypercube of anti-basis u and signature R (Lemma A.8), the total size of ψ-labelings is at most $O(2^d n)$ too.

The notion of *milestone* intervenes in the definition of labels ψ. We consider two vertices u, v such that $u \in I(v_0, v)$. Milestones are defined recursively.

▶ **Definition C.11** (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: $\Pi(u, v) = \{u\} \cup \Pi(u^+, v)$.*

The milestones are the successive anti-bases of the hypercubes formed by the vertices and ladder sets traversed from u to v. Both vertices u and v are contained in $\Pi(u, v)$. The first milestone is u, the second is the anti-basis u^+ of the hypercube with basis u and signature $L_{u,v}$. The third one is the anti-basis u^{++} of the hypercube with basis u^+ and signature $L_{u^+,v}$, etc. All milestones are metrically between u and v: $\Pi(u, v) \subseteq I(u, v)$.

▶ **Definition C.12** (Penultimate milestone $\overline{\pi}(u, v)$). *We say that the milestone in $\Pi(u, v)$ different from v but the closest to it is called the penultimate milestone. We denote it by $\overline{\pi}(u, v)$. Furthermore, we denote by $\overline{L}_{u,v}$ the* anti-ladder set *of u, v,* i.e. *the Θ-classes of the hypercube with basis $\overline{\pi}(u, v)$ and anti-basis v.*

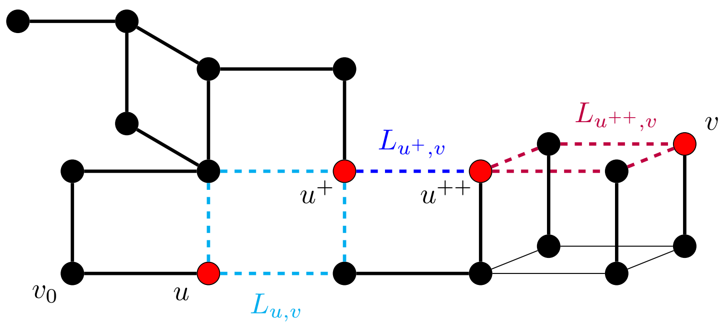


Figure 14 A pair u, v with $u \in I(v_0, v)$ and its milestones $\Pi(u, v)$ in red.

Figure 14 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 have $\overline{\pi}(u, v) = u^{++}$ and $\overline{L}_{u,v} = L_{u^{++},v}$ is drawn in purple.

Let R be a POF ingoing to some vertex u and u^- be the basis of the hypercube with anti-basis u and signature R. Label $\psi(u, R)$ intuitively represents the maximum distance of a shortest path arriving to vertex u from "direction" R.

▶ **Definition C.13** (Labels ψ [14]). *The label $\psi(u, R)$ is the maximum distance $d(u, v)$ we can obtain with a vertex v satisfying the following properties:*

- $m = m(u, v, v_0) \neq u$,
- *the anti-ladder set of m, v is R: $\overline{L}_{m,v} = R$.*

Equivalently, vertex u^- is the penultimate milestone of pair m, u: $u^- = \overline{\pi}(m, u)$.

As for the computation of labels φ, there is an induction process to determine all $\psi(u, R)$. As the base case, suppose that $u^- = v_0$. The largest distance $d(u, v)$ we can obtain with a vertex v such that $v_0 \in I(u, v)$ consists in considering the opposite $\mathsf{op}_{v_0}(R)$ of R which is outgoing from v_0. Hence, $\psi(u, R) = |R| + \varphi(v_0, \mathsf{op}_{v_0}(R))$.

For the induction step, we distinguish two cases. In the first one, assume that $m(u, v, v_0) = u^-$ - equivalently, $\Pi(m, u) = \Pi(u^-, u) = \{u^-, u\}$. A shortest (u, v)-path is the concatenation of the shortest (u, u^-)-path of length $|R|$ with a shortest (u^-, v)-path, and $u^- \in I(v_0, v)$. The largest distance $d(u, v)$ we can have, as for the base case, is $\psi(u, R) = |R| + \varphi(u^-, \mathsf{op}_{u^-}(R))$.

In the second case, $m \neq u^-$, an inductive formula allows us to obtain $\psi(u, R)$. A consequence of Lemma C.6 is that, for two consecutive milestones in $\Pi(u, v)$, say u and u^+ w.l.o.g, then $L_{u^+,v}$ is $L_{u,v}$-parallel. This observation, applied to the penultimate milestone, provides us with the following theorem.

▶ **Lemma C.14** (Inductive formula for labels ψ [14]). *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:*

(i) vertex v is the penultimate milestone of (u, w): $\overline{\pi}(u, w) = v$,

(ii) the milestones of (u, w) are the milestones of (u, v) with w: $\Pi(u, w) = \Pi(u, v) \cup \{w\}$,

(iii) the POF L is $\overline{L}_{u,v}$-parallel.

Set $\Pi(m, u)$ admits at least three milestones: m, u^-, and u. Let R^- be the POF ingoing to u^- which is the ladder set (but also the signature) of (i) the milestone just before u^- and (ii) u^-. According to Lemma C.14, vertex u^- is the penultimate milestone of (m, u) if and only if $R^- \cup \{E_j\}$ is not a POF, for each $E_j \in R$. For this reason, value $\psi(u, R)$ can be expressed as:

$$\psi(u, R) = \max_{\substack{R^- \text{ POF ingoing to } u^- \\ \forall E_j \in R, R^- \cup \{E_j\} \text{ not POF}}} (|R| + \psi(u^-, R^-)) \tag{5}$$

Our algorithm consists in taking the maximum value between the two cases. The number of pairs $((u, R), (u^-, R^-))$ which satisfy the condition described in Equation (5) is at most $2^{2d}n$: it is identical to the one presented for φ-labelings.

▶ **Theorem C.15** (Computation of labels ψ [14]). *There is a combinatorial algorithm which determines all labels $\psi(u, R)$ in $\tilde{O}(2^{2d}n)$.*

C.1.4 Better time complexity for all eccentricities

The computation of all labels $\varphi(u, L)$, $\mathsf{op}_u(L)$ and $\psi(u, R)$ gives an algorithm which determines all eccentricities. Indeed, each eccentricity $\mathsf{ecc}(u)$ is a function of certain labels φ and ψ. Let v be a vertex in G such that $\mathsf{ecc}(u) = d(u, v)$. If $m = m(u, v, v_0) = u$, then $u \in I(v_0, v)$ and value $d(u, v)$ is given by a label $\varphi(u, L)$. Otherwise, if $m \neq u$, let u^- be the penultimate milestone in $\Pi(m, u)$ and R be the classes of the hypercube with basis u^- and anti-basis u. The eccentricity of u is given by a label $\psi(u, R)$. Conversely, each $\varphi(u, L)$ and $\psi(u, R)$ is the distance between u and another vertex by definition. Therefore, we have:

$$\mathsf{ecc}(u) = \max \left\{ \max_{\substack{L \text{ POF} \\ \text{outgoing from } u}} \varphi(u, L), \max_{\substack{R \text{ POF} \\ \text{ingoing to } u}} \psi(u, R) \right\} \tag{6}$$

In other words, the eccentricity of u is the maximum label φ or ψ centered at u. We can conclude with the main result of this subsection: the eccentricities of any median graph can be determined in linear time multiplied by a simple exponential function $2^{O(d)}$ of the dimension d.

▶ **Theorem C.16** (All eccentricities in $\tilde{O}(2^{2d}n)$-time for median graphs)**.** *There is a combinatorial algorithm computing the list of all eccentricities of a median graph G in time $\tilde{O}(2^{2d}n)$.*

Proof. We simply determine the labels φ, op and ψ with the algorithms mentioned in Theorem C.7, C.10, and C.15. Thanks to the time improvement obtained for the computation of opposite labels, the overall running time to compute the labels is only $\tilde{O}(2^{2d}n)$. Then, for each vertex u, Equation (6) guides us to obtain its eccentricity. We must take the maximum over all labels $\varphi(u, L)$ - they are at most N_u - and all labels $\psi(u, R)$ - they are at most 2^d. As $\sum_u N_u \leq 2^d n$, the execution time of this operation on all vertices is $\tilde{O}(2^d n)$. Therefore, it does not overpass the time complexity needed to determine the labels. ◀

C.2 Proofs of Section 4.2

▶ **Lemma C.17.** *Let G be a median graph. For every $1 \leq i \leq q$, let $v \in V(H_i')$ be arbitrary, and let $v^* \in \partial H_i''$ be its gate. Then, $\mathsf{ecc}(v) = \max\{\mathsf{ecc}_{H_i'}(v), d(v, v^*) + \mathsf{ecc}_{H_i''}(v^*)\}$.*

Proof. We have $\mathsf{ecc}(v) = \mathsf{ecc}_G(v) = \max\{d(u, v) \mid u \in V(H_i')\} \cup \{d(w, v) \mid w \in V(H_i'')\}$. Since H_i' is convex, we have $\max\{d(u, v) \mid u \in V(H_i')\} = \mathsf{ecc}_{H_i'}(v)$. In the same way, since H_i'' is gated (and so, convex), we have $\max\{d(w, v) \mid w \in V(H_i'')\} = d(v, v^*) + \max\{d(v^*, w) \mid w \in V(H_i'')\} = d(v, v^*) + \mathsf{ecc}_{H_i''}(v^*)$. ◀

▶ **Lemma C.18.** *Let H and G be median graphs. If H is an induced subgraph of G then, every Θ-class of H is contained in a Θ-class of G.*

Proof. Every square of H is also a square of G. In particular, two edges of H are in relation Θ_0 if and only if, as edges of G, they are also in relation Θ_0. Since the Θ-classes of H (resp., of G) are the transitive closure of its relation Θ_0, it follows that every Θ-class of H must be contained in a Θ-class of G. ◀

▶ **Lemma C.19.** *Let H and G be median graphs, and let $E_1, E_2, \ldots, E_q$ denote the Θ-classes of G. If H is an isometric subgraph of G then, the Θ-classes of H are exactly the nonempty subsets among $E_i \cap E(H)$, for $1 \leq i \leq q$.*

Proof. It is known [41] that two edges uv, xy of G are in the same Θ-class if and only if $d_G(u, x) + d_G(v, y) \neq d_G(u, y) + d_G(v, x)$. In particular, since H is isometric in G, two edges of H are in the same Θ-class of H if and only if they are in the same Θ-class of G. ◀

▶ **Lemma C.20.** *Let G be a median graph, and let $D := \max\{|E_i| \mid 1 \leq i \leq q\}$ be the maximum cardinality of a Θ-class of G. Then, $d = \dim(G) \leq \lfloor \log D \rfloor + 1$, and this bound is sharp.*

Proof. Let Q_d be an induced d-dimensional hypercube of G. Every Θ-class of Q_d contains 2^{d-1} edges (namely, for some fixed choice of $j \in \{1, \ldots, d\}$, it contains all edges uv such that the binary representations of u and v only differ in position j). By Lemma C.18, any Θ-class of Q_d must be contained in a Θ-class of G. As a result, $2^{d-1} \leq D$. The bound is reached if $G = Q_d$. ◀

D Appendix for Section 5

D.1 Reach centrality

In this subsection, we propose a linear FPT algorithm, parameterized by d, dedicated to the computation of all reach centralities of a median graph G. The *reach centrality* $\mathrm{RC}(u)$ of a vertex u is a parameter related to the length of shortest paths on which vertex u lies. The farther a vertex u is from the two extremities of a shortest path traversing it, the larger the reach centrality of u is. This notion originally inspired some efficient routing strategies on road networks [24]. The relationship between reach centrality and the well-known metric parameters has been studied: Abboud *et al.* [1] proved that determining the diameter and the reach centrality are equivalent under subcubic reductions. The formal definition of $\mathrm{RC}(u)$ follows.

$$\mathrm{RC}(u) = \max_{u \in I(s,t)} \min\{d(s,u), d(u,t)\} \tag{7}$$

In Theorem C.16, we showed that all eccentricities of a median graph are functions of the labelings φ, op, and ψ. Here, a similar result is established for the reach centralities.

We begin with a first observation which will be useful to state the dependence of RC on the labels already computed. Under a certain orthogonality condition (the R-parallelism of L), for any pair of POFs R and L respectively ingoing in and outgoing from $u \in V$, there are two vertices s, t such that $d(u,s) = \varphi(u,L)$, $d(u,t) = \psi(u,R)$ and $u \in I(s,t)$.

▶ **Theorem D.1.** *Let $u \in V$, R be a POF ingoing into u and L a POF outgoing from u such that L is R-parallel. There exists a pair (s,t) of vertices satisfying the following properties:*

- *Vertex u belongs to interval $I(v_0, s)$ and $L_{u,s} = L$,*
- *The median $m = m(s,t,v_0)$ is different from u and $\overline{L}_{m,u} = R$.*
- *The distance $d(u,s)$ and $d(u,t)$ are given by the labels: $d(u,s) = \varphi(u,L)$ and $d(u,t) = \psi(u,R)$.*
- *Vertex u belongs to the interval $I(s,t)$.*

Proof. Let s be a vertex such that $u \in I(v_0,s)$, $L_{u,s} = L$ and $d(u,s) = \varphi(u,L)$. Let t be a vertex such that $\overline{L}_{m,u} = R$ for $m = m(u,t,v_0)$, and $d(u,t) = \psi(u,R)$. By definition of labels φ and ψ, such vertices exist. At this moment, the three first bullets are verified. To show the fourth one, $u \in I(s,t)$, we prove that the signatures $\sigma_{u,s}$ and $\sigma_{u,t}$ are disjoint. Assume there is a Θ-class $E_i \in \sigma_{u,s} \cap \sigma_{u,t}$.

Claim 1: $u \in H_i'$. As $E_i \in \sigma_{u,s}$, any shortest (u,s)-path contains an edge of E_i. Let (u',v') be one of these edges of E_i which is as close as possible from u. Vertex u' - the endpoint of this edge closer to u - belongs to $\partial H_i'$. Moreover, there is a shortest (u,s)-path $P_{u,s}$ passing through (u',v') because v' is the gate of u in H_i''. As $u \in I(v_0,s)$ and $u' \in I(u,s)$, then $u \in I(v_0,u')$. We know that both v_0 and u' belong to H_i': by convexity of halfspaces, $u \in H_i'$.

Claim 2: $u \in H_i''$. We have $E_i \in \sigma_{u,t}$: we prove that E_i is necessarily in $\sigma_{m,u}$ and not in $\sigma_{m,t}$. The class E_i cannot form a POF if we add it to R. We already know it if $E_i \in L$. We prove that: if it was the case for a class $E_i \in \sigma_{u,s} \backslash L$, then it would imply the orthogonality of R and all Θ-classes of L, a contradiction. Indeed, let z be the vertex such that $\mathcal{E}^-(z) = R \cup \{E_i\}$. We have $v', z \in \partial H_i''$. As v' is the gate of u in H_i'', there is a shortest (u,z)-path passing through (u',v'). We denote by $\partial H_R''$ the intersection of all $\partial H_j''$ for all $E_j \in R$. As $u, z \in \partial H_R''$, which is convex, all vertices metrically between u and z belong to

$\partial H''_R$, in particular v'. The ladder set of u, v' is the same as u, s because v' does not belong to the hypercube $Q_{u,L}$ of basis u and signature L: $L_{u,v'} = L$. In brief, $Q_{u,L} \subsetneq I(u, v')$. So, all Θ-classes of R are adjacent to the vertices of $Q_{u,L}$, a contradiction as $R \cup L$ is not a POF.

We know now that there is a class $E_j \in R$ such that E_i and E_j are parallel. Thus, $H''_i \subsetneq H''_j$. Suppose, by way of contradiction, that $E_i \in \sigma_{m,t}$. Vertex t is in H''_i, so it is also in H''_j. As $E_j \in R \subseteq \sigma_{m,u}$, then $m \in H'_j$: m and t are not in the same halfspace of E_j. In other words, $E_j \in \sigma_{m,t}$. This is a contradiction because E_j is both in $\sigma_{m,u}$ and $\sigma_{m,t}$ while m is metrically between u and t. Any shortest (u,t)-path should pass through two edges of E_j, which is not possible from Lemma A.2. Finally, $E_i \in \sigma_{m,u}$ and $u \in H''_i$.

Both Claims 1 and 2 yield a contradiction: the signature sets $\sigma_{u,s}$ and $\sigma_{u,t}$ must be disjoint. Therefore, $u \in I(s,t)$. ◀

We present now an algorithm which determines, for any vertex $u \in V$, a label $\chi(u)$. The vertices can be considered in any arbitrary order. The objective is to obtain, at the end of the execution, $\chi(u) = \mathrm{RC}(u)$, for any vertex u. To start, we fix all $\chi(u)$ equal to 0. Let $\xleftarrow{\max}$ be the operator which modify the left-hand side variable with the maximum between itself and the right-hand side one. Formally, for $a \in \mathbb{N}$, $\chi(u) \xleftarrow{\max} a$ is equivalent to $\chi(u) \leftarrow \max\{\chi(u), a\}$. Given a vertex $u \in V$, we proceed in three steps.

Step 1: Reach when u is the median of s, t, v_0. This step amounts to determining the reach centrality of u if we restrict ourselves to pairs s, t such that $u = m(s,t,v_0)$. Given a vertex s such that $u \in I(v_0, s)$, the extremity t maximizing $d(u,t)$ such that $u = m(s,t,v_0)$ is at distance $\varphi(u, \mathsf{op}(L_{u,s}))$, according to Lemma C.9. If $d(u,s) \le \varphi(u, L_{u,s}) \le \varphi(u, \mathsf{op}_u(L_{u,s}))$, then distance $d(u,s)$ has no influence on $\mathrm{RC}(u)$ as another candidate - any vertex at distance $\varphi(u, L_{u,s})$ from u - overpasses it. Moreover, if $d(u,s) \le \varphi(u, \mathsf{op}_u(L_{u,s})) \le \varphi(u, L_{u,s})$, then $d(u,s)$ also cannot be equal to $\mathrm{RC}(u)$ because $\varphi(u, \mathsf{op}_u(L_{u,s}))$ overpasses it and will count, according to Lemma C.9 and the fact that $\varphi(u, L_{u,s})$ is greater than it. Eventually, if $d(u,s) > \varphi(u, \mathsf{op}_u(L_{u,s}))$, it does not count as we cannot form a pair (s,t) such that $u \in I(s,t)$ and $d(s,u) \le d(u,t)$. In summary, the only values that have to be taken into account for the reach centrality when u is a median are the φ-labelings $\varphi(u, L)$. We modify label $\chi(u)$ according to these observations.

For any POF L outgoing from u, if $\varphi(u, L) < \varphi(u, \mathsf{op}_u(L))$, then modify the label $\chi(u) \xleftarrow{\max} \varphi(u, L)$, otherwise do nothing.

Step 2: Reach when $u \neq m = m(s,t,v_0)$ but is a milestone of m, s. Let L be the ladder set of u, s; $L = L_{u,s}$ and R the anti-ladder set of m, u; $R = \overline{L}_{m,u}$. According to Lemma C.14, L is R-parallel. Theorem D.1 intervenes: there is a pair of vertices s^*, t^* such that $u \in I(s,t)$, $L_{u,s^*} = L$, $\overline{L}_{m^*,u} = R$, $d(u, s^*) = \varphi(u, L)$ and $d(u, t^*) = \psi(u, R)$, where $m^* = m(s^*, t^*, v_0)$. As $\varphi(u, L) \ge d(u,s)$ and $\psi(u, R) \ge d(u,t)$, the reach centrality of u in this step can be written as a function of only φ, ψ-labelings. For example, if a POF L admits an anti-ladder set R such that $\varphi(u, L) < \psi(u, R)$, then it must be taken into account for the computation of $\chi(u)$.

For any pair L, R of POFs respectively outgoing from and ingoing to u such that L is R-parallel: if $\varphi(u, L) < \psi(u, R)$, we modify the label $\chi(u) \xleftarrow{\max} \varphi(u, L)$. Otherwise, we set $\chi(u) \xleftarrow{\max} \psi(u, R)$.

Step 3: Reach when $u \neq m = m(s,t,v_0)$ and is not a milestone of m, s. Let u' be the milestone of $\Pi(m,s)$ and its u'' its successor in $\Pi(m,s)$ such that u belongs to the hypercube of basis u', anti-basis u'' and, hence, signature $L = L_{u',u''} = \sigma_{u',u''}$. We distinguish two cases.

- *Case 1: u' is the median of s, t, v_0.* In this case, the distance $d(u,s)$ is less than $\varphi(u', L) - d(u', u)$. Moreover, the distance $d(u,t)$ is less than $d(u', u) + \varphi(u', \mathsf{op}_{u'}(L))$. So, the reach centrality can be expressed only as a function of labels φ, ψ, and the distance $d(u', u)$.

For any POF L outgoing from some $u' \in V$ such that u belongs to the hypercube of basis u' and signature L, if $\varphi(u', L) - d(u', u) < d(u', u) + \varphi(u', \mathsf{op}_{u'}(L))$, then we modify the label $\chi(u) \xleftarrow{\max} \varphi(u', L) - d(u', u)$. Otherwise, we set $\chi(u) \xleftarrow{\max} d(u', u) + \varphi(u', \mathsf{op}_{u'}(L))$.

- *Case 2: u' is not the median of s, t, v_0.* Let R be the anti-ladder set of m, u'. Theorem D.1 implies the existence of a pair of vertices s^*, t^* with the same (anti-)ladder sets L and R than u' with s, t and such that $d(u', s^*) = \varphi(u', L)$ and $d(u', t^*) = \psi(u', R)$. Furthermore, Lemma C.3 ensures us that a shortest path between u' and s^* can be prefixed with the Θ-classes of L in any ordering. As a consequence, there is a shortest (u', s^*)-path containing u. As $\varphi(u', L) - d(u', u) \geq d(u, s)$ and $\psi(u', R) + d(u', u) \geq d(u, t)$, the reach centrality of u in this step can be written only as a function of φ, ψ-labelings and distance $d(u, u')$.

For any pair L, R of POFs respectively outgoing from and ingoing to some $u' \in V$ such that L is R-parallel: enumerate all vertices u belonging to the hypercube of basis u' and signature L. For each of them, if $\varphi(u', L) - d(u', u) < \psi(u, R) + d(u', u)$, we modify the label $\chi(u) \xleftarrow{\max} \varphi(u', L) - d(u', u)$. Otherwise, we set $\chi(u) \xleftarrow{\max} \psi(u', R) + d(u', u)$.

Pseudocode. Algorithm 2 provides us with the pseudocode of this procedure. The steps corresponding to the updates of $\chi(u)$ are mentioned as comments, surrounded by symbol #.

Algorithm 2 Computation of labels χ

Input: Median graph G, weight function $C : V \to \mathbb{N}$, labels $\varphi, \mathsf{op}, \psi$, $\varphi, \mathsf{op}, \psi$.
Output: Labels $\chi(u)$ for any vertex $u \in V$.
Initialize $\chi(u) \leftarrow 0$ for any vertex u;
for *every pair (u, L) where L is a POF outgoing from u* **do**
  if $\varphi(u, L) < \varphi(u, \mathsf{op}_u(L))$ **then**
    $\chi(u) \xleftarrow{\max} \varphi(u, L)$; # Step 1 #
  endif
  for *every vertex u^* belonging to the hypercube of basis u and signature L* **do**
    $\chi(u^*) \xleftarrow{\max} \min\{\varphi(u, L) - d(u, u^*), \varphi(u, \mathsf{op}_u(L)) + d(u, u^*)\}$; # Step 3-1 #
  endfor
  for *every POF R ingoing into u such that L is R-parallel* **do**
    $\chi(u) \xleftarrow{\max} \min\{\varphi(u, L), \psi(u, R)\}$ # Step 2 # ;
    for *every vertex u^* belonging to the hypercube of basis u and signature L* **do**
      $\chi(u^*) \xleftarrow{\max} \min\{\varphi(u, L) - d(u, u^*), \psi(u, R) + d(u, u^*)\}$; # Step 3-2 #
    endfor
  endfor
endfor

The computation of all labels φ, op, and ψ is a necessary preprocessing of this algorithm. We remind that they can be obtained in $\tilde{O}(2^{2d} n)$. Steps 1, 2 and 3 cover all possible configurations of triplet u, s, t such that $\mathrm{RC}(u) = \min\{d(s, u), d(s, t)\}$. Indeed, either u is the median of s, t, v_0 (Step 1) or not. If not, it is either a milestone of at least one pair among $(m(s, t, v_0), s)$ and $(m(s, t, v_0), t)$ (Step 2), or not (Step 3). In each situation, both distances $d(s, u)$ and $d(u, t)$ are upper-bounded in function of some label values. Conversely, these upper bounds correspond to the distance between u and certain vertices s^*, t^*, such that $u \in I(s^*, t^*)$. Hence, $\mathrm{RC}(u)$ can be expressed as a function of labelings φ, op, and ψ, as described in Algorithm 2.

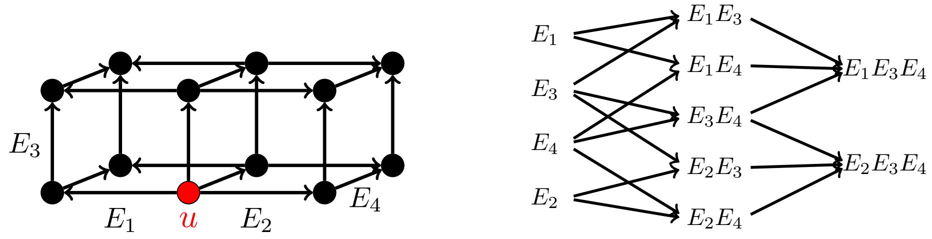
▶ **Theorem D.2.** *There is a combinatorial algorithm computing all reach centralities $RC(u)$ of a median graph in $\tilde{O}(2^{3d}n)$.*

Proof. The correctness of Algorithm 2 is now clear. We focus on its runtime. The most expensive part corresponds to Step 3, Case 2 (line 14). Indeed, we must enumerate all triplets (u, L, R): we know they are at most $2^{2d}n$. For each of them, we list all vertices lying on the hypercube of basis u and signature L, which contains potentially 2^d vertices. The total number of 4-uplets (u, L, R, u^*) considered in line 14 of Algorithm 2 is thus at most $2^{3d}n$. ◀

D.2 Proofs of Section 5.2

▶ **Theorem D.3.** *Assume graph G has at most $\tilde{O}(f(d,n)n)$ MOPs, $f(d,n) = o(2^d)$. There is a combinatorial algorithm computing all labels φ, op, and ψ in $\tilde{O}(2^d f(d,n)n)$.*

Proof. Let $u \in V$, N_u be the number of hypercubes with basis u. We remind that $\sum_{u \in V} N_u \leq 2^d n$. We begin with the definition of a DAG H_u^{L} for hypercubes. It is called the *ladder Hasse diagram* of u. Its vertex set is made up of all pairs (u, L) of hypercubes with basis u, in other words, L is outgoing from u. There is an arc $(u, L') \to (u, L)$ if $L' \subsetneq L$ and $|L'| = |L| - 1$. All diagrams H_u^{L}, $u \in V$, can be constructed in time $\tilde{O}(2^d n)$. Indeed, all hypercubes can be enumerated in $\tilde{O}(2^d n)$ with a BFS (Lemma A.10). Then, it suffices, for each (u, L), to consider the at most d subsets $L' \subsetneq L$ differing from one element from L and connect (u, L') to (u, L). In this way, we obtain a directed graph where its connected components are the diagrams H_u^{L}. An example of DAG H_u^{L} follows.



(a) Example of star graph G_u **(b)** DAG H_u^{L} representing its hypercubes

Figure 15 Ladder Hasse diagram H_u^{L}

Thanks to the ladder Hasse diagram, computing a list of the MOPs in $\tilde{O}(2^d n)$ is straightforward as they are exactly the leaves of the DAGs H_u^{L}.

From now on, we divide the proof into three steps, each one correspond to the computation of a certain type of labels. We begin with the labels $\varphi(u, L)$.

Computation of φ-labelings. We present a new procedure to compute the labels $\varphi(u, L)$. A key distinction with the former method should be mentioned: instead of determining labels $\varphi(u, L)$ only, we also compute another type of labels, denoted by $\varphi_{\subseteq}(u, L)$. These *maximal ladder* labels must contain in fact the maximum over all labels $\varphi(u, L')$ such that $L' \subseteq L$. In brief, $\varphi_{\subseteq}(u, L) = \max_{L' \subseteq L} \varphi(u, L')$.

We compute these two label functions recursively. First, we propose a recursive procedure to obtain $\varphi(u, L)$ based on the enumeration of MOPs. We list all triplets (L, u^+, L^+) such that (u^+, L^+) is a MOP and L is ingoing into u^+. We denote by u the basis of the hypercube with anti-basis u^+ and signature L. The base case is the same as in Section C.1.1. If (u, L) is such that either no edge is leaving the anti-basis u^+ or all Θ-classes outgoing from u^+ are orthogonal to L, then fix $\varphi(u, L) = |L|$. We pursue with the inductive step. We denote by $L_{\perp}^+$

the maximal L-parallel subset of L^+, *i.e.* the set containing exactly the Θ-classes E_i of L^+ for which $L \cup \{E_i\}$ is not a POF. As $L^+_\perp \subseteq L^+$, set $L^+_\perp$ is a POF. The computation of $L^+_\perp$ implies a logarithmic extra cost of $O(d)$ for each MOP (u^+, L^+). We write the MOP-equivalent formula of Equation (4).

$$\varphi(u,L) = \max_{\substack{(u^+,L^+) \text{ MOP, } L \text{ ingoing into } u \\ \forall E_j \in L^+_\perp,\ L\cup\{E_j\} \text{ not POF}}} (|L| + \varphi_\subseteq(u^+, L^+_\perp)). \tag{8}$$

We explain why Equation (8) is correct. According to Equation (4), $\varphi(u,L)$ is $|L|$ plus the maximum over all $\varphi(u^+,L^+_*)$ - not necessarily MOPs - such that L^+_* is L-parallel. Assume (u^+,L^+_*) is not a MOP: there is a MOP (u^+,L^+), where $L^+_* \subsetneq L^+$. As $L^+_\perp$ is the maximal L-parallel subset of L^+, we have $L^+_* \subseteq L^+_\perp$. So, $\varphi(u^+,L^+_*) \leq \varphi_\subseteq(u^+,L^+_\perp)$ by definition. Conversely, value $\varphi_\subseteq(u^+,L^+_\perp)$ counts in the computation of $\varphi(u,L)$ because all subsets of $L^+_\perp$ are L-parallel. Therefore, it suffices to consider the MOPs (u^+,L^+) with their maximal subset $L^+_\perp$ instead of all hypercubes (u^+,L^+_*).

Second, we explain how the labels $\varphi_\subseteq$ are deduced from the values $\varphi(u,L)$. Assume that for a given vertex u, all $\varphi(u,L)$, L outgoing from u, have been determined recursively, thanks to the base case or Equation (8). We deduce all $\varphi_\subseteq(u,L)$ with the ladder Hasse diagram structure H^{L}_u. We proceed inductively. The base case concerns singleton POFs: $\varphi(u,\{E_i\}) = \varphi_\subseteq(u,\{E_i\})$. Then, we describe the induction step. We modify the value $\varphi_\subseteq(u,L)$ by comparing $\varphi(u,L)$ with all $\varphi_\subseteq(u,L')$, where $L' \subsetneq L$, $|L'| = |L|-1$. In other words, we initialize $\varphi_\subseteq(u,L)$ as $\varphi(u,L)$ and, for each arc $(u,L') \to (u,L)$ of H^{L}_u, we execute $\varphi_\subseteq(u,L) \xleftarrow{\max} \varphi_\subseteq(u,L')$. Concretely, we transfer the $\varphi_\subseteq$-labelings from the roots to the leaves of the diagram H^{L}_u.

To compute labels $\varphi(u,L)$, the enumeration of MOPs is needed to apply Equation (8). For each MOP (u^+,L^+), we must consider all POFs L ingoing into u, which gives a total of at most $\tilde{O}(2^d f(d,n)n)$ triplets (L,u^+,L^+), as the number of MOPs is $\tilde{O}(f(d,n)n)$. The logarithmic extra costs do not increase this runtime. To compute labels $\varphi_\subseteq(u,L)$, our induction is based on the structure of all diagrams H^{L}_u. The total size (number of arcs) of the DAG H^{L}_u is $\tilde{O}(N_u)$ as each element has at most d parents. Therefore, the execution time needed to determine all $\varphi_\subseteq(u,L)$ does not exceed $\tilde{O}(2^d n)$. In summary, the entire procedure to compute $\varphi, \varphi_\subseteq$-labelings is in $\tilde{O}(2^d f(d,n)n)$.

Computation of op*-labelings.* We already know that all labels $\mathsf{op}_u(L)$ can be determined in time $\tilde{O}(2^d n)$, according to Theorem C.10.

Computation of ψ*-labelings.* Our inductive procedure to determine all labels $\psi(u,R)$ based on the MOP structure is in fact very close to the one produced for φ-labelings.

We define the *anti-ladder Hasse diagram* H^{AL}_u of u. Its vertex set is made up of all pairs (u,R) of hypercubes with anti-basis u and a POF R ingoing into u. There is an arc $(u',R') \to (u,R)$ if both hypercubes (defined by their anti-basis and signature) have the same basis, $R' \supsetneq R$ and $|R'| = |R|+1$. As for diagrams H^{L}_u, all DAGs H^{AL}_u can be constructed in time $\tilde{O}(2^d n)$ with a standard BFS.

As for φ-labelings, we define two label functions which will be computed jointly. However, the description is a bit trickier. We compute not only labels $\psi(u,R)$ but also the new ones $\psi_\supseteq(u,R)$. Contrary to $\varphi_\subseteq$-labelings, value $\psi_\supseteq(u,R)$ is not so easy to define. To understand, we remind the inductive process to compute labels $\psi(u,R)$ in Section C.1.3.

Remember that there are two cases. Let u^- be the basis of (u,R). First, value $\psi(u,R)$ can be given by a distance $d(u,v)$ such that $m(u,v,v_0) = u^-$. In this case, $\psi(u,R) = |R| + \varphi(u^-, \mathsf{op}_{u^-}(R))$. In the new procedure, we initialize all $\psi(u,R)$ with this value. Second, we may have $m(u,v,v_0) \neq u^-$. In this case, value $\psi(u,R)$ is given by the recursive formula in Equation (5). We can now define labels $\psi_\supseteq$. Let R^- be a POF ingoing into u^- and assume that (u^-,R) is a MOP. Let $R_\perp$ be the set containing exactly the Θ-classes E_i of R such that R is R^--parallel. We denote by $u_\perp$ the anti-basis of the hypercube with basis u^- and signature $R_\perp$. Then, value $|R|+\psi(u^-,R^-)$, which counts originally in the computation of $\psi(u,R)$ (Equation (5)), will count only for the computation of $\psi_\supseteq(u_\perp,R_\perp)$. More formally,

$$\psi_\supseteq(u_\perp,R_\perp) = \max_{\substack{R^- \text{ POF ingoing to } u^-,\\ \exists (u^-,R) \text{ MOP with } R_\perp \text{ max subset of } R\\ \text{such that } \forall E_j \in R_\perp, R^- \cup \{E_j\} \text{ not POF}}} (|R|+\psi(u^-,R^-)) \tag{9}$$

Observe that certain pairs (u,R) may not admit a value $\psi_\supseteq(u,R)$ according to this definition. In this case, we simply fix $\psi_\supseteq(u,R)=0$.

Now, we show that value $\psi(u,R)$, in the case $m \neq u^-$, is exactly the maximum over all $\psi_\supseteq(u',R')$ such that both hypercubes (defined by anti-basis and signature) have the same basis and $R \subseteq R'$. On one hand, if (u^-,R) is not a MOP, there is a MOP (u^-,R^*), where $R \subsetneq R^*$. Let R^- be a POF ingoing into u^- such that R is R^--parallel. Let $R^*_\perp$ be the maximal R^--parallel subset of R^*. Set $R^*_\perp$ is a POF and $R \subseteq R^*_\perp$ by definition. Assume $\psi(u,R) = |R|+\psi(u^-,R^-)$: this value is counted in $\psi_\supseteq(u^*_\perp,R^*_\perp)$ but not in $\psi_\supseteq(u,R)$ if $R \neq R^*_\perp$. On the other hand, if some $R' \supseteq R$ is R^--parallel, then R also does. So, we have:

$$\psi(u,R) = \max \left\{ |R| + \varphi(u^-, \mathsf{op}_{u^-}(R)), \max_{\substack{(u',R') \text{ same basis as } (u,R)\\ R \subseteq R'}} \psi_\supseteq(u',R') \right\}. \tag{10}$$

Given a basis u^-, we compute all nonnegative values $\psi_\supseteq(u,R)$ such that u^- is the basis of the hypercube with anti-basis u and signature R. To do so, we enumerate all triplets (R^-,u^-,R) such that R^- is ingoing into u^- and (u^-,R) is a MOP. We compute the maximal R^--parallel subset $R_\perp$ of R and apply Equation (9). As for φ-labelings, it consist in an enumeration scheme in $\tilde{O}(2^d f(d,n)n)$.

Then, we deduce labels $\psi(u,R)$ of the hypercubes with basis u^-. We use the anti-ladder Hasse diagram H_u^{AL}. If (u^-,R) is a MOP, then (u,R) is a root of H_u^{AL}, and we fix $\psi(u,R)$ as the maximum between its initial value (case $m(u,v,v_0)=u^-$) and $\psi_\supseteq(u,R)$: it corresponds to Equation (10) when R is maximal. Otherwise, $\psi(u,R)$ can be computed from Equation (10) in function of $\psi_\supseteq(u',R')$ where (u',R') is a parent of (u,R) in H_u^{AL}. The time cost of this step is at most $\tilde{O}(2^d n)$, due to the size of DAG H_u^{AL}. In summary, we obtain the same global running time than for labels $\varphi(u,L)$, which is $\tilde{O}(2^d f(d,n)n)$. ◀

▶ **Lemma D.4** (MOPs as subsets of maximal POFs). *Let (u,L) be a MOP and $p(u,L) = (L,L^*)$. Then, L^* is a maximal POF.*

Proof. Let u^+ be the anti-basis of (u,L). Assume, by way of contradiction, that L^* is not maximal. There is a Θ-class E_h such that $L^{**} = L^* \cup \{E_h\}$ is a POF. Let u' be the vertex such that $\mathcal{E}^-(u') = L^{**}$.

For every $E_i \in L^*$, vertices u^+ and u' belongs to $\partial H_i''$. The intersection of these boundaries, that we denote by ∂H^*, is convex. Therefore, all vertices metrically between u^+ and u' belong to ∂H^*. For this reason, there is necessarily an edge e incident to u^+ whose second endpoint is also in ∂H^*. Let $e=(u^+,w)$ be this edge and we denote by E_j its Θ-class. If e is ingoing into u^+ with the v_0-orientation, then we have a contradiction, since

$p(u, L) = (L, L^*) \neq (L, L^{**})$. We know that e is outgoing from u^+. We remind that E_j is orthogonal to any Θ-class of L^*. By successive applications of Lemma A.5, we show that for any vertex of the hypercube of anti-basis u^+ and signature L^*, among them u, there is an edge of E_j outgoing from it. Here comes the contradiction: L is not a maximal POF outgoing from u as $L \cup \{E_j\}$ is a POF ($L \subseteq L^*$). ◀

▶ **Corollary D.5** (MOPs as subsets of maximal cliques in $G^{\#}$). *Let $G^{\#}$ be the crossing graph of G and $\mathcal{C}^{\#}_{max}$ be the set of maximal cliques of $G^{\#}$. The number of MOPs in G is at most* $\sum_{C \in \mathcal{C}^{\#}_{max}} 2^{|C|}$.

Proof. We begin with the proof that application p is injective. Let us consider a pair $(L, L^*) = p(u, L)$. There is a unique vertex u^+ such that $\mathcal{E}^- = L^*$. Necessarily, if $(u, L) = p^{-1}(L, L^*)$, then vertex u must be the basis of the hypercube with signature L and anti-basis u^+, which is unique.

Now, application p is restricted to MOPs only. We know it is injective and that, for any MOP (u, L), set L^* of $p(u, L) = (L, L^*)$ is a maximal POF. Concretely, the number of MOPs is at most the following value: for each maximal POF, add the number of its nonempty subsets (which is 2 power its cardinality minus 1).

We remind that a POF of G corresponds to an induced clique (can be empty) of its crossing graph $G^{\#}$. Naturally, a maximal clique of $G^{\#}$ represents a maximal POF. It follows that the number of MOPs of G is at most $\sum_{C \in \mathcal{C}^{\#}_{\text{max}}} (2^{|C|} - 1) \leq \sum_{C \in \mathcal{C}^{\#}_{\text{max}}} 2^{|C|}$. ◀

▶ **Theorem D.6.** *Let H be a graph with $|V(H)| = q$, clique number at most d, which maximizes $r(H)$. If H is not complete multipartite, there is another graph H' with $|V(H')| = q$, clique number at most d, such that $r(H') = r(H)$ and $\mathrm{Trp}(H') < \mathrm{Trp}(H)$.*

Proof. If H is not complete multipartite, there is a pair u, v of vertices such that $(u, v) \notin E$ and $N(u) \neq N(v)$. Let N_u (resp. N_v) be the number of cliques of H containing u (resp. v). For $x \in \{u, v\}$, we denote by R_x the following value: $R_x = \sum_{\substack{C \in \mathcal{C}_{\text{max}}(H) \\ x \in C}} 2^{|C|}$. For the remainder, we fix $\Delta R = R_v - R_u$ and $\Delta N = N_v - N_u$. We define two graphs $H_{u \to v}$ and $H_{v \to u}$, and compare their maximal clique ratio with the initial graph H.

Graph $H_{u \to v}$ is obtained from H by removing u and adding a copy v' of v such that $N(v') = N(v)$. We have $r(H_{u \to v}) = \frac{R[H] + \Delta R}{N[H] + \Delta N}$, as u and v cannot belong to the same clique. Conversely, graph $H_{v \to u}$ is obtained from H by removing v and adding a copy u' of u such that $N(u') = N(u)$. We have $r(H_{v \to u}) = \frac{R[H] - \Delta R}{N[H] - \Delta N}$. Both $H_{u \to v}$ and $H_{v \to u}$ do not increase the clique number of H. One can check that if $\frac{R[H] + \Delta R}{N[H] + \Delta N} \leq \frac{R[H]}{N[H]}$, then $\frac{R[H] - \Delta R}{N[H] - \Delta N} \geq \frac{R[H]}{N[H]}$ and vice-versa. If the inequality is strict, then we have a contradiction since H is supposed to maximize $r(H)$ for graphs with q vertices and clique number at most d. The only possibility we have is $r(H) = r(H_{u \to v}) = r(H_{v \to u})$.

We prove that either $H_{u \to v}$ or $H_{v \to u}$ has less triplets with only two adjacent vertices than H. Let Trp_u (resp. Trp_v) be the number of triplets of H containing u and not v (resp. v and not u). Let $\mathrm{Trp}_{u/v}$ (resp. $\mathrm{Trp}_{v/u}$) the number of triplets of H containing both u and v, where u is the isolated vertex (resp. v is the isolated vertex). We have

$$\mathrm{Trp}(H_{u \to v}) = \mathrm{Trp}(H) + \mathrm{Trp}_v - \mathrm{Trp}_u - \mathrm{Trp}_{u/v} - \mathrm{Trp}_{v/u},$$
$$\mathrm{Trp}(H_{v \to u}) = \mathrm{Trp}(H) + \mathrm{Trp}_u - \mathrm{Trp}_v - \mathrm{Trp}_{u/v} - \mathrm{Trp}_{v/u}.$$

At least one of this values is smaller than $\mathrm{Trp}(H)$, otherwise $\mathrm{Trp}_{u/v} = \mathrm{Trp}_{v/u} = 0$, which is equivalent to saying $N(u) = N(v)$, a contradiction. ◀

▶ **Theorem D.7.** *Turán graphs $T(q,d)$ maximize the maximal clique ratio for graphs with q vertices and clique number d.*

Proof. We consider a complete multipartite graph H with q vertices and clique number d. The vertex set of H can be partitioned into d independent sets A_i: $V(H) = \bigcup_1^d A_i$. Let $\alpha_i = |A_i|$. Assume, w.l.o.g, that the sizes of sets A_i are increasing, *i.e.* $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_d$. We remind that $\sum_1^d \alpha_i = q$. Turán graphs are the graphs such that $\alpha_d - \alpha_1 \leq 1$.

Assume that, on graph H, $\alpha_d - \alpha_1 \geq 2$. We prove adding a vertex to A_1 and removing another one from A_d increases the maximal clique ratio, without changing q or d. Let H' denote the graph after this transformation. All maximal cliques in H have size d, so $R[H] = 2^d \prod_1^d \alpha_i$. The number $N[H]$ of (not necessarily maximal) cliques of H is expressed as: $N[H] = \sum_{\mathcal{J} \subseteq \{1,\ldots,d\}} \prod_{j \in \mathcal{J}} \alpha_j$. Values $R[H']$ and $N[H']$ can be deduced by replacing respectively α_1 and α_d by $\alpha_1 + 1$ and $\alpha_d - 1$. We assess the quotient between $r(H')$ and $r(H)$. Certain details of our calculations are omitted to keep the paper readable, we restrict ourselves to the main steps of the reasoning.

$$\frac{r(H')}{r(H)} = \frac{\frac{R[H']}{R[H]}}{\frac{N[H']}{N[H]}} = \frac{\frac{\alpha_1+1}{\alpha_1}\frac{\alpha_d-1}{\alpha_d}}{1 + \frac{\alpha_d-\alpha_1-1}{(\alpha_1+1)(\alpha_d+1)}} = \left(1 + \frac{1}{\alpha_1(\alpha_1+2)}\right)\left(1 - \frac{1}{\alpha_d^2}\right) > 1.$$

Indeed, as $\alpha_d - \alpha_1 \geq 2$, one can check that $\left(1 + \frac{1}{\alpha_1(\alpha_1+2)}\right)\left(1 - \frac{1}{(\alpha_1+2)^2}\right)$ is greater than 1 for any value of α_1. ◀

▶ **Corollary D.8.** *The number of MOPs in a median graph is $O(f(d,n)n)$, where $f(d,n) = \left(2.\frac{2^{\frac{\log n}{d}}-1}{2^{\frac{\log n}{d}}}\right)^d$.*

Proof. According to Corollary 5.6, the number of MOPs of a median graph G is upper-bounded by $r(G^{\#})n$. Furthermore, Theorem D.7 states that the balanced complete multipartite graphs maximize $r(H)$. Hence,

$$r(G^{\#}) \leq \frac{2^d(\frac{q}{d})^d}{\sum_{j=0}^d \binom{d}{j}(\frac{q}{d})^j} \leq \frac{2^d(\frac{q}{d})^d}{(1+\frac{q}{d})^d} \tag{11}$$

The number of cliques of $G^{\#}$ is exactly the number of POFs of G. Therefore, $n = |V(G)| = (1+\frac{q}{d})^d$. We fix $a = \frac{d}{\log n}$, we have $0 < a \leq 1$. We obtain that $2^{\frac{d}{a}} = (1+\frac{q}{d})^d$ and, thus, $\frac{q}{d} = 2^{\frac{1}{a}} - 1$. Finally, we inject this equality into Equation (11). ◀