



**HAL**  
open science

# Distance problems within Helly graphs and k-Helly graphs

Guillaume Ducoffe

► **To cite this version:**

Guillaume Ducoffe. Distance problems within Helly graphs and k-Helly graphs. *Theoretical Computer Science*, 2023, 946, pp.113690. 10.1016/j.tcs.2023.113690 . hal-03942267

**HAL Id: hal-03942267**

**<https://hal.science/hal-03942267>**

Submitted on 17 Jan 2023

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

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

# Distance problems within Helly graphs and $k$ -Helly graphs

Guillaume Ducoffe<sup>1,2</sup>

<sup>1</sup>National Institute for Research and Development in Informatics, Romania

<sup>2</sup>University of Bucharest, Romania

## Abstract

The *ball hypergraph* of a graph  $G$  is the family of balls of all possible centers and radii in  $G$ . Balls in a subfamily are  *$k$ -wise intersecting* if the intersection of any  $k$  balls in the subfamily is always nonempty. The *Helly number* of a ball hypergraph is the least integer  $k$  greater than one such that every subfamily of  $k$ -wise intersecting balls has a nonempty common intersection. A graph is  *$k$ -Helly* (or *Helly*, if  $k = 2$ ) if its ball hypergraph has Helly number at most  $k$ . We prove that a central vertex and *all* the medians in an  $n$ -vertex  $m$ -edge Helly graph can be computed w.h.p. in  $\tilde{O}(m\sqrt{n})$  time. Both results extend to a broader setting where we assign a nonnegative cost to the vertices. For any fixed  $k$ , we also present an  $\tilde{O}(m\sqrt{kn})$ -time randomized algorithm for radius computation within  $k$ -Helly graphs. If we relax the definition of Helly number (for what is sometimes called an “almost Helly-type” property in the literature), then our approach leads to an approximation algorithm for computing the radius with an additive one-sided error of at most some constant.

**Keywords:** Radius and Median computations; Helly graphs; Graph algorithms.

## 1 Introduction

For any undefined graph terminology, see [11, 27, 42]. All graphs considered are simple (*i.e.*, they have no loops and no multiple edges), connected and with unit-length edges. We study the complexity of computing several distance parameters within some increasing hierarchy of graph classes. Specifically, let  $G = (V, E)$  be an undirected graph. The distance between two vertices  $u$  and  $v$ , denoted by  $dist(u, v)$ , is equal to the minimum number of edges on a  $uv$ -path in  $G$ . The eccentricity of a vertex  $v$  is defined as  $e(v) = \max_{u \in V} dist(u, v)$ . The total distance of a vertex  $v$  is defined as  $TD(v) = \sum_{u \in V} dist(u, v)$ . We focus on two fundamental *facility location* problems:

- The CENTER problem asks for a vertex  $v$  such that  $e(v)$  is minimized (*a.k.a.*, central vertex).
- The MEDIAN problem asks for a vertex  $v$  such that  $TD(v)$  is minimized (*a.k.a.*, median).

Facility location problems are about finding optimal locations, w.r.t. some distance and cost criteria, for opening new facilities such as schools or hospitals. Due to their practical relevance, such problems have been widely studied in mathematical programming, both on metric spaces and graphs. However, our motivation for studying these two problems also comes from their importance in the fine-grained complexity study of polynomial-time solvable problems. Indeed, we observe that both the CENTER and MEDIAN problems can be solved in  $\mathcal{O}(n^2)$  time, if we are given the distance

matrix of the graph. Computing the distance matrix, or equivalently solving the All-Pairs Shortest-Paths problem (APSP), can be done in  $\mathcal{O}(nm)$  time, or in  $\mathcal{O}(n^{\omega+\epsilon})$  time, for every positive  $\epsilon$ , where  $\omega < 2.273$  denotes the square matrix multiplication exponent [51]. If we further assume the input graph to be sparse (or of arbitrary density, but assuming  $\omega = 2$ ), then this is in  $\mathcal{O}(n^{2+o(1)})$  time, which is optimal. By our previous observation, these quadratic-time upper bounds also apply to the CENTER and MEDIAN problems. However, for a long time it was open whether these upper bounds are asymptotically tight. In 2016, Abboud et al. proved that it was indeed the case if we assume the so-called ‘‘Hitting Set Conjecture’’, that implies the Orthogonal Vectors Conjecture [1]. Specifically, their conjecture implies that for any  $\epsilon > 0$ , neither the CENTER nor the MEDIAN problems can be solved in  $\mathcal{O}(n^{2-\epsilon})$  time on  $n$ -vertex  $\tilde{\mathcal{O}}(n)$ -edge graphs. In particular, this rules out the existence of an  $\tilde{\mathcal{O}}(n^a m^b)$ -time algorithm, for every positive  $a, b$  such that  $a + b < 2$ . We stress that this conditional lower bound also holds for *radius* computation (*i.e.*, the seemingly weaker problem of computing the minimum eccentricity, but not necessarily a corresponding central vertex).

We follow a long line of work in Algorithmic Graph Theory, that consists in improving the complexity of distance problems on some special graph classes. Some of these prior works have considered center and/or median computations in linear time on subclasses of bounded treewidth graphs [39, 41], bounded clique-width graphs [26, 33, 34], planar graphs [22, 38, 39] and chordal graphs [25, 28, 52, 47]. For general chordal graphs, we can compute a central vertex in linear time [20, 21]. More recently, almost linear-time algorithms for the CENTER and MEDIAN problems on general graphs of bounded treewidth were presented in [1, 14, 16]. Subquadratic-time algorithms for these problems on general planar graphs were presented in [15, 40]. It is worth mentioning that all the aforementioned graph classes are characterized by forbidden induced subgraphs or forbidden minors, that is unlike the graph classes considered in the present work.

The ball of center  $v$  and radius  $r$  is the set  $N^r[v] = \{u \in V \mid \text{dist}(u, v) \leq r\}$ . The *ball hypergraph* of  $G$ , denoted by  $\mathcal{B}(G)$ , is the family of the balls of all possible centers and radii in  $G$ . That is,  $\mathcal{B}(G) = \{N^r[v] \mid v \in V, r \geq 0\}$ . Several interesting graph classes can be defined via the properties of their ball hypergraphs. For instance, the dually chordal graphs are exactly the graphs of which the ball hypergraph is a dual hypertree [13]. There exists a linear-time algorithm for computing the radius of a dually chordal graph [12]. More recently, we proved in [37], with co-authors, that there exists a parameterized subquadratic-time algorithm for computing the radius, within all the graph classes whose ball hypergraphs have VC-dimension upper bounded by a constant. As a particular case, we got a truly subquadratic-time algorithm for radius computation within all the proper minor-closed classes of graphs. This work makes a new step toward finding a more general property of ball hypergraphs that we could still exploit for a fast radius computation.

Specifically, balls in a subfamily are *k-wise intersecting* if the intersection of any  $k$  balls in the subfamily is always nonempty. A graph is *k-Helly* if every sub-family of  $k$ -wise intersecting balls has a nonempty common intersection. The 2-Helly graphs have already received some attention in the literature, under the different names of *Helly graphs* (that we use in this paper), *disk-Helly graphs* [8] or *absolute retracts* [6]. We refer to Fig. 1 for an example of Helly graph. Notably, the interval graphs, the strongly chordal graphs and the dually chordal graphs are all subclasses of the Helly graphs [13]. In fact, a celebrated result in Metric Graph Theory is that *every* graph is an isometric (distance preserving) subgraph of some Helly graph [35, 43]. Therefore, unlike most of the special graph classes studied in the literature, the class of Helly graphs *cannot* be defined via a set of forbidden patterns such as (induced) subgraphs, minors or bounded expansion. Note that in general, a bounded VC-dimension does *not* imply a bounded Helly number, but it does

imply a slightly weaker *fractional Helly property* [46]. We refer to the survey [4], that has devoted a full section to the properties of Helly graphs. See also [5, 6, 7, 8, 19, 28, 29, 30, 36, 48, 49] for other properties, and, *e.g.*, [6, 28, 45] for improved polynomial-time recognition algorithms for the Helly graphs. For  $k \geq 3$ , we stress that we can decide in  $n^{\mathcal{O}(k)}$  time whether an  $n$ -vertex graph is  $k$ -Helly, that is polynomial for every fixed  $k$ ; indeed, a graph is  $k$ -Helly if and only if for every  $(k + 1)$ -subset  $S$  of vertices, the balls intersecting  $S$  in at least  $k$  vertices have a nonempty common intersection [10]. The running times of these recognition algorithms are superquadratic, but we do *not* need to execute them before we can apply our own algorithms in this paper.

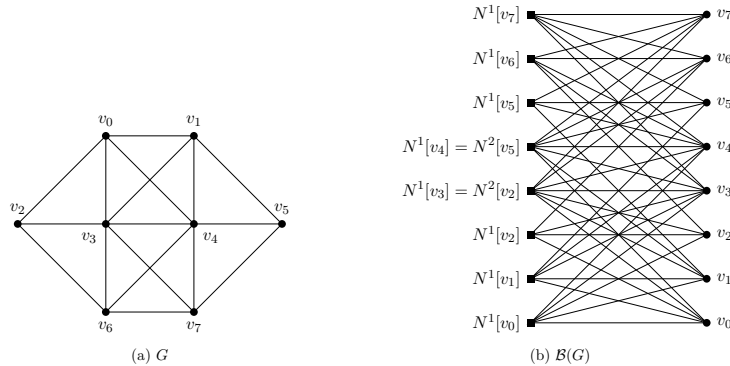


Figure 1: A Helly graph (left) and its ball hypergraph (right). The balls  $N^0[v_i]$ , for every  $i$  such that  $0 \leq i \leq 7$  (reduced to a single vertex), and the ball containing all vertices are not represented.

Recently, with Dragan, we initiated the fine-grained complexity study of polynomial-time solvable distance problems within Helly graphs [36]. Specifically, we proved that the radius of an  $n$ -vertex  $m$ -edge Helly graph can be computed w.h.p. in  $\tilde{O}(m\sqrt{n})$  time. Our result followed from a known reduction to the computation of a diametral pair, *i.e.*, a pair of vertices whose distance in the graph is maximum. Indeed, for a Helly graph of diameter  $D$ , the radius is exactly  $\lceil D/2 \rceil$  [28]. Unfortunately, we failed in also computing a central vertex with this approach.

## Our Contributions

**Helly graphs.** We further refine the framework in [36], in order to solve the CENTER and MEDIAN problems in truly subquadratic time within the Helly graphs.

- Our first result in this paper is an algorithm for computing a central vertex in a Helly graph, that runs in  $\tilde{O}(m\sqrt{n})$  time w.h.p. (Theorem 1). For that, we use the nice property that the eccentricity function of a Helly graph is *unimodal*: every local minimum is a global minimum [28]. The latter suggests the following natural *local search* strategy in order to compute a central vertex: start from some vertex with a hopefully small eccentricity, then repeatedly find a neighbour with a better eccentricity until a local minimum is found. Obvious drawbacks to this approach are: the time needed for each step of the local search, and the total number of steps. In order to overcome the first issue, we use and further refine some “gated” properties of the sets of small diameter in a Helly graph. We handle the second issue with a classical random sampling technique, that allows us to find, amongst  $\tilde{O}(\sqrt{n})$

candidates, a start vertex which ensures a sufficiently small (sublinear) total number of steps. For other recent applications of this sampling technique to distance problems, see [2, 50].

- Our second main result is a similar algorithm for computing the median in a Helly graph, that also runs in  $\tilde{\mathcal{O}}(m\sqrt{n})$  time w.h.p. (Theorem 2). Indeed, since the total distance function of a Helly graph is also unimodal [5], we can use the same local search strategy as before in order to compute a median. For a Helly graph, the subset of all medians is a clique [5]. By using this nice structural property, we can compute all the medians in one more step of the local search. We note that very recently, a similar local search algorithm was presented in [9] in order to compute all the medians in linear time within median graphs. The authors in [9] achieved a better running time than ours by exploiting the special properties of median graphs. In particular, they derived, after some linear-time preprocessing, a constant-time algorithm for comparing the total distances of two adjacent vertices in a median graph.

Our approach still works for a more general version of the CENTER and MEDIAN problems: where we assign costs to the vertices. We stress that in [36], we used different properties of unimodal functions in order to compute a diametral pair. In a follow-up paper [31], based on the techniques developed here and in [36], we proved that we can also compute the center of a Helly graph in truly subquadratic time (*i.e.*, the set of all its central vertices). Note that for a Helly graph, computing the center is equivalent to computing all the eccentricities [28].

**$k$ -Helly graphs.** Our third main result in the paper is a randomized algorithm for computing the radius of a  $k$ -Helly graph, which runs w.h.p. in  $\tilde{\mathcal{O}}(m\sqrt{kn})$  time (Theorem 3). In particular, our result implies that the hard instances for radius computation must have a quasi linear Helly number. Furthermore, if we relax the Helly property, as described next, then we obtain an approximation algorithm for computing the radius. Specifically, let us call a graph  $(k, \alpha)$ -Helly if every family of  $k$ -wise intersecting balls has a nonempty common intersection when we increase the radius of each ball by  $\alpha$ . For the special case  $k = 2$ , this relaxed property has been studied under the names of Helly-gap [32] or coarse Helly property [17]. Our algorithm outputs the radius with some additive one-sided error in  $\mathcal{O}(\alpha)$ .

Closest to our work on the  $k$ -Helly graphs is a very general framework by Amenta [3]. Namely, she proved that on set families of constant Helly number, deciding whether these sets have a nonempty common intersection could be rephrased, under some mild conditions, as a so-called Generalized Linear Program (GLP) with a bounded combinatorial dimension. The GLP's are a broad generalization of linear programming, for which the prototypical problem is the computation of a central point in Euclidean spaces of bounded dimension. If a GLP has a bounded combinatorial dimension, then we can solve it in randomized linear time, by using Clarkson's algorithm [24]. However, the latter relies on two primitives, namely basis computation and violation testing. We do not know how to implement these two basic operations in constant time, nor even in sublinear time, for the special case of graphs. Therefore, we could not use Amenta's framework in order to prove our results directly, although we stayed close to the spirit of the latter. Our algorithm requires  $k$  to be given, and it fails in also computing a central vertex. Roughly, this is because we fail in satisfying another technical condition from Amenta's framework, namely the so-called "Unique Minimum Condition". We left open whether, for any fixed integer  $k \geq 3$ , a central vertex can be computed in truly subquadratic time within the  $k$ -Helly graphs.

## 2 Facility Location problems on Helly graphs

In what follows, let  $c : V \rightarrow \mathbb{R}^+$  be a cost assignment to the vertices of some graph  $G = (V, E)$ . We define the  $c$ -eccentricity and the total  $c$ -distance of a vertex  $v$  as  $e_c(v) = \max_{u \in V} c(u) \cdot \text{dist}(u, v)$  and  $TD_c(v) = \sum_{u \in V} c(u) \cdot \text{dist}(u, v)$ , respectively. The main results of this section are truly subquadratic algorithms, for the CENTER and MEDIAN problems, respectively, on Helly graphs (Theorems 1 and 2). More generally, for any function  $c$  that assigns nonnegative costs to the vertices, we can compute a vertex minimizing its  $c$ -eccentricity, resp. *all* vertices minimizing their total  $c$ -distance. For that, we first need to introduce several intermediate results on Helly graphs and unimodal functions.

**Unimodal functions.** Let  $G = (V, E)$  be a graph. In what follows, we denote by  $N(v)$  the set of neighbours of a vertex  $v$ . Furthermore, let  $f : V \rightarrow \mathbb{R}^+$ . A local minimum for  $f$  is a vertex  $v$  such that, for every  $u \in N(v)$ , we have  $f(u) \geq f(v)$ . If furthermore  $f(v) = \min_{w \in V} f(w)$ , then  $v$  is a global minimum for  $f$ . We denote by  $\arg \min(f)$  the set of all the global minima. Finally, we recall that  $f$  is called unimodal if every local minimum is also a global minimum.

**Lemma 1** (Chapter 2 in [28]). *If  $G = (V, E)$  is a Helly graph, then for any nonnegative cost function  $c$ , the function  $e_c : V \rightarrow \mathbb{R}^+$  that maps every vertex  $v$  to its  $c$ -eccentricity is unimodal.*

**Lemma 2** (Corollary 5.1 in [5]). *If  $G = (V, E)$  is a Helly graph, then for any nonnegative cost function  $c$ , the function  $TD_c : V \rightarrow \mathbb{R}^+$  that maps every vertex  $v$  to its total  $c$ -distance is unimodal. Moreover, the subset of all medians of  $G$  is a clique.*

Throughout the paper, by a *random subset*  $U(p)$  we mean a vertex subset in which every vertex was added independently at random with probability  $p$ . We say that a path  $P = [v_0, v_1, \dots, v_\ell]$  is  $f$ -monotone if we have  $f(v_i) > f(v_{i+1})$  for every  $i$  with  $0 \leq i < \ell$ . Our starting point for this section is the following observation:

**Lemma 3.** *Let  $G = (V, E)$  be a graph and  $f : V \rightarrow \mathbb{R}^+$  be unimodal. Being given a random subset  $U(p)$ , if  $u$  is any vertex that minimizes  $f(u)$  within  $U(p)$ , then w.h.p. every  $f$ -monotone path between  $u$  and  $\arg \min(f)$  has length at most in  $\tilde{O}(p^{-1})$ .*

*Proof.* For every vertex  $v$  we fix a longest  $f$ -monotone path  $P_v$  between  $v$  and  $\arg \min(f)$ . This choice is independent of  $U(p)$ . Let  $V_p$  be the set of vertices  $v$  such that  $|E(P_v)| > 1 + 2p^{-1} \log n$ . Let  $\mathcal{E}_p$  be the event that there exists a vertex  $v$  of  $V_p$  such that  $(V(P_v) \setminus \{v\}) \cap U(p) = \emptyset$ . We have:

$$\begin{aligned} \Pr[\mathcal{E}_p] &\leq \sum_{v \in V_p} \Pr[(V(P_v) \setminus \{v\}) \cap U(p) = \emptyset] = \sum_{v \in V_p} (1-p)^{|V(P_v)|-1} \\ &\leq \sum_{v \in V_p} (1-p)^{\frac{2 \log n}{p}} \leq \sum_{v \in V_p} n^{-2} \leq n^{-1}. \end{aligned}$$

Now, let us assume the existence of some vertex  $u$  minimizing  $f(u)$  within  $U(p)$  such that the length of  $P_u$  is larger than  $1 + 2p^{-1} \log n$ . By minimality of  $f(u)$  we have  $(V(P_u) \setminus \{u\}) \cap U(p) = \emptyset$ . Then, the probability for such a vertex  $u$  to exist is at most  $\Pr[\mathcal{E}_p]$ , that is at most  $1/n$ .  $\square$

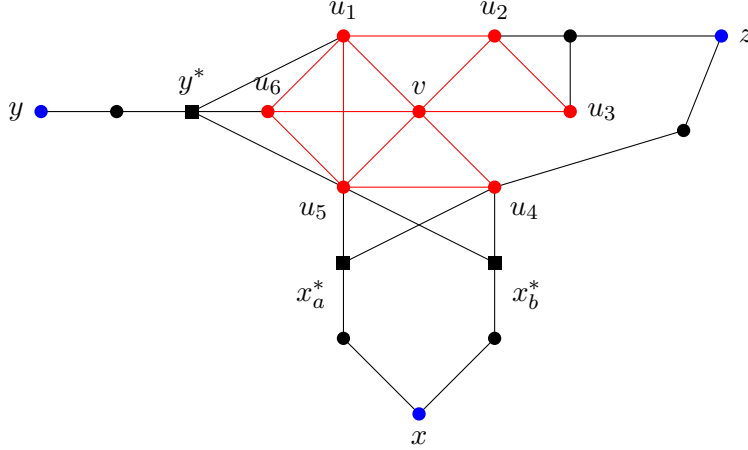


Figure 2: For  $S = N[v]$ , vertex  $x$  has two gates  $x_a^*, x_b^*$  and one pseudo-gate  $u_5$ . Vertex  $y$  has a gate  $y^*$  in  $S$  but no pseudo-gate. Vertex  $z$  has no gate in  $S$ .

**Gates and Pseudo-gates.** Given a graph  $G$  and a vertex  $v$ , we recall that the ball of center  $v$  and radius  $r$  is denoted by  $N^r[v]$ . For any vertex subset  $S$  we define  $dist(v, S) = \min_{s \in S} dist(v, s)$ . The metric projection of  $v$  on  $S$  is defined as  $Proj(v, S) = \{s \in S \mid dist(v, s) = dist(v, S)\}$ . The (weak) diameter of a set  $S$  is equal to  $diam(S) = \max_{x, y \in S} dist(x, y)$ . The following notions of gates (Def. 1) and pseudo-gates (Def. 2) are an important cornerstone of our algorithmic approach in the paper (see also Fig. 2):

**Definition 1.** For every vertex  $v$  not in a subset  $S$ , a *gate* of  $v$  in  $S$  is any vertex  $v^* \in N(S)$  that is on a shortest  $vx$ -path for every vertex  $x$  in  $Proj(v, S)$ . Formally,  $v^* \in N^{dist(v, S)-1}[v] \cap \bigcap \{N(x) \mid x \in Proj(v, S)\}$ .

We denote by  $g_S(v)$  an arbitrary gate of vertex  $v$  in  $S$ , if there exists at least one such vertex, or simply by  $g(v)$  if  $S$  is clear from the context.

A vertex may have no gate, one gate, or many gates in  $S$ . In particular, a necessary condition for a vertex  $v$  to have a gate in  $S$  is that the weak diameter of its metric projection on  $S$  is at most two. This condition is not sufficient. For instance, in an odd cycle of length at least five, if  $v$  denotes an arbitrary vertex and the subset  $S$  contains the two furthest vertices from  $v$ , that are adjacent, then  $Proj(v, S) = S$  whereas there is no common neighbour to both vertices in  $S$ .

The notion of gate can be traced back to early works in [44] on chordal graphs, from which it was deduced in [18] that in the special case where  $S$  is a clique in a chordal graph every vertex  $v$  not in  $S$  has a gate. This nice property was later used in [20] for the design of a linear-time algorithm for computing a central vertex in a chordal graph. Roughly, being given a clique  $S$  and a gate  $g(v)$  for every vertex  $v$  not in  $S$ , after some linear-time preprocessing we may compute the eccentricities and the total distances of *every* vertex  $s$  in  $S$  simply by enumerating the gates in their respective neighbourhoods  $N(s)$ . For that, we use that for every vertex  $v$  not in  $S$ ,  $dist(v, s) = dist(v, S)$  if and only if  $s \in Proj(v, S)$ , that is if and only if  $g(v) \in N(s)$ . Otherwise, since  $S$  is a clique,  $dist(v, s) = dist(v, S) + 1$ . We use a similar approach in Lemma 7 below. However, since we need to consider neighbourhoods, not necessarily cliques, we need the additional concept of pseudo-gate.

**Definition 2.** For every vertex  $v$  not in a subset  $S$ , a *pseudo-gate* of  $v$  in  $S$  is any vertex  $v' \in N[S]$  such that  $\text{dist}(v, v') \leq \text{dist}(v, S)$ ,  $N^{\text{dist}(v, S)+1}[v] \cap S \subseteq N[v']$ , and  $v'$  is either equal or adjacent to some gate  $g_S(v)$ .

We denote by  $pg_S(v)$  an arbitrary pseudo-gate of vertex  $v$  in  $S$ , if there exists at least one such vertex, or simply by  $pg(v)$  if  $S$  is clear from the context.

In particular, a pseudo-gate may only exist if  $v$  has a gate in  $S$ . However, a vertex may have a gate and no pseudo-gate. For instance, let  $S_0$  be an arbitrary subset in a graph  $G_0$  and let  $G$  be obtained from  $G_0$  by adding for every vertex  $s$  in  $S_0$  a degree-one vertex  $s'$  only adjacent to  $s$ . Let  $S = \{s, s' \mid s \in S_0\}$ . If a vertex  $v \notin S$  has a gate in  $S_0$ , then it is also a gate in  $S$ . However, if  $|Proj(v, S)| > 1$ , then  $v$  has no pseudo-gate because  $N^{\text{dist}(v, S)+1}[v]$  contains all degree-one vertices  $s' \in S \setminus S_0$  that are adjacent to  $Proj(v, S)$ . Note that we allow a pseudo-gate to be either in  $S$  or in  $N(S)$ , whereas a gate can only be in  $N(S)$ .

The concept of pseudo-gate was first introduced in [36] where it was proved that every vertex has a pseudo-gate if the weak diameter of  $S$  is at most two and the graph  $G$  is  $C_4$ -free Helly. To better understand the role played by pseudo-gates in our algorithm, let us consider a subset  $S$ , a vertex  $v \notin S$ , and a vertex  $s \in S \setminus Proj(v, S)$ . If moreover  $S$  is a clique, then we know that  $\text{dist}(v, s) = \text{dist}(v, S) + 1$ . However, if  $S$  is a neighbourhood, then we only know that  $\text{dist}(v, s) \in \{\text{dist}(v, S)+1, \text{dist}(v, S)+2\}$ . We can use a pseudo-gate of  $v$ , if any, in order to determine  $\text{dist}(v, s)$ .

Being given a Helly graph  $G$ , and a subset  $S$  of weak diameter at most two, we use in our Lemma 7 below that every vertex  $v \notin S$  has both a gate and a pseudo-gate in  $S$ . The existence of a gate was first proved in [36], namely:

**Lemma 4** (Lemma 10 in [36]). *Let  $G$  be a Helly graph and  $S$  be a subset of weak diameter at most two. Then, every vertex  $v \notin S$  has a gate in  $S$ .*

The existence of a pseudo-gate was only proved in [36] for the  $C_4$ -free Helly graphs. We now generalize [36, Lemma 19] to all the Helly graphs.

**Lemma 5.** *Let  $G$  be a Helly graph and let  $S$  be a subset of weak diameter at most two. Then, every vertex  $v \notin S$  has a pseudo-gate in  $S$ .*

*Proof.* For convenience, let us write  $\text{dist}(v, S) = r - 1$ . Let  $\mathcal{F} = \{N^{r-1}[v]\} \cup \{N[x] \mid x \in S \cap N^r[v]\}$ . Since the balls in  $\mathcal{F}$  pairwise intersect, by the Helly property there exists a vertex  $v' \in N^{r-1}[v] \cap \bigcap \{N[x] \mid x \in S \cap N^r[v]\}$ . We prove in what follows that vertex  $v'$  is a pseudo-gate of vertex  $v$  in  $S$ . First,  $\text{dist}(v, v') \leq \text{dist}(v, S)$  because  $v' \in N^{r-1}[v]$  and  $\text{dist}(v, S) = r - 1$ . Second,  $N^{\text{dist}(v, S)+1}[v] \cap S \subseteq N[v']$  because  $\text{dist}(v, S) + 1 = r$  and  $v' \in \bigcap \{N[x] \mid x \in S \cap N^r[v]\}$ . It remains to prove the existence of a gate  $v^*$  of vertex  $v$  such that  $v^* \in N[v']$ . For that, consider the subfamily  $\mathcal{F}' = \{N^{r-2}[v], N[v']\} \cup \{N[z] \mid z \in Proj(v, S)\}$ . We claim that the balls in  $\mathcal{F}'$  pairwise intersect. Indeed, since we have  $v' \in N^{r-1}[v]$  and  $r - 1 > 0$ , we get  $N^{r-2}[v] \cap N[v'] \neq \emptyset$ . In the same way, for every vertex  $z$  in  $Proj(v, S)$ ,  $N^{r-2}[v] \cap N[z] \neq \emptyset$  because  $\text{dist}(v, z) = \text{dist}(v, S) = r - 1$ . For every vertex  $z$  in  $Proj(v, S)$ , we also have  $v' \in N[z]$  because  $z \in S \cap N^r[v]$  and by construction  $v' \in \bigcap \{N[x] \mid x \in S \cap N^r[v]\}$ . In particular,  $N[v'] \cap N[z] \neq \emptyset$ . Finally, for all vertices  $z$  and  $z'$  in  $Proj(v, S)$ ,  $N[z] \cap N[z'] \neq \emptyset$  because  $\text{dist}(z, z') \leq \text{diam}(S) \leq 2$ . So, by the Helly property, there exists a vertex  $v^* \in N^{r-2}[v] \cap \{N[z] \mid z \in Proj(v, S) \cup \{v'\}\}$ . By construction,  $v^* \in N[v']$ . Furthermore, for every vertex  $z$  of  $Proj(v, S)$ , vertex  $v^*$  must be on a shortest  $vz$ -path because we have  $v^* \in N^{r-2}[v] \cap N[z]$  and  $\text{dist}(v, z) = r - 1$ . Then,  $v^* \in N^{\text{dist}(v, S)-1}[v] \cap \bigcap \{N(z) \mid z \in Proj(v, S)\}$ . Therefore,  $v^*$  is a gate of vertex  $v$ .  $\square$



As first observed in [20] (see also Remark 1 in [36]), for every subset  $S$  in an arbitrary graph  $G$ , we can compute in linear time a gate  $g_S(v)$  for all vertices  $v \notin S$  for which such a vertex exists. We can also compute in linear time a pseudo-gate  $pg_S(v)$  for all vertices  $v \notin S$  for which such a vertex exists (see Remark 3 in [36]). For completeness of the paper, we present in what follows a linear-time algorithm for both computations. In what follows, if  $\mathcal{P}$  denotes a subset of  $\mathbb{N} \times \mathbb{N}$ , then we call a pair  $(a, b)$  *lexicographically maximum* within  $\mathcal{P}$  if for every other pair  $(a', b')$  of  $\mathcal{P}$ , either  $a' < a$ , or  $a' = a$  and  $b' \leq b$ .

**Lemma 6** (Section 3 in [20], Remarks 1 and 3 in [36]). *Let  $S$  be an arbitrary nonempty subset in some graph  $G = (V, E)$ . In total  $\mathcal{O}(m)$  time, we can map every vertex  $v \notin S$  to vertices  $v^*, v'$  such that  $(|N(v^*) \cap S|, |N[v'] \cap S|)$  is lexicographically maximum within the set of all vertex pairs  $u, w$  such that  $u \in N^{dist(v, S)-1}[v]$ ,  $w \in N[u]$ . Furthermore, two properties hold:*

1. *if  $v$  has a gate in  $S$ , then  $v^*$  is a gate of  $v$ ;*
2. *if  $v$  has a pseudo-gate in  $S$ , then  $v'$  is a pseudo-gate of  $v$ .*

*Proof.* Let  $G'_S$  be obtained from  $G \setminus S$  by adding a fresh new vertex  $x_S$  and the edges  $x_S v$  for every vertex  $v \in N_G(S)$ . We run a BFS on  $G'_S$  starting from  $x_S$ . It can be done in  $\mathcal{O}(m)$  time. In doing so, we compute  $dist(v, S)$  for every vertex  $v \notin S$ . We also compute, in total  $\mathcal{O}(m)$  time, the values  $d_S(v) = |N[v] \cap S|$  for every vertex  $v \in N[S]$ . Then, we consider the vertices  $v \notin S$  sequentially, by nondecreasing distance to  $S$ .

- Case  $dist(v, S) = 1$ . We set  $v^* = v$  and we pick a vertex  $v'$  maximizing  $d_S(v')$  within  $N[v]$ .
- Case  $dist(v, S) > 1$ . Let  $y_1, y_2, \dots, y_t$  denote all the neighbours of  $v$  that are at distance  $dist(v, S) - 1$  to  $S$ . We denote by  $y_i^*, y'_i$ , for every  $i$  such that  $1 \leq i \leq t$ , the pair of vertices to which we mapped vertex  $y_i$ . We pick a pair  $(v^*, v') \in \{(y_1^*, y'_1), \dots, (y_t^*, y'_t)\}$  so that  $(d_S(v^*), d_S(v'))$  is lexicographically maximum.

In both cases, we only need to scan  $N[v]$  at most twice, that can be done in  $\mathcal{O}(|N[v]|)$  time. Therefore, the total running time is in  $\mathcal{O}(m)$ .

We prove the correctness of the algorithm for all vertices  $v \notin S$  by induction on  $dist(v, S)$ . This is straightforward if  $dist(v, S) = 1$ . Thus, from now on we assume that  $dist(v, S) > 1$ . Let  $(u, w)$  be a pair of vertices such that  $u \in N^{dist(v, S)-1}[v]$ ,  $w \in N[u]$  and  $(d_S(u), d_S(w))$  is lexicographically maximized. In order to prove that the algorithm is correct for vertex  $v$ , it suffices to prove that  $d_S(u) = d_S(v^*)$  and  $d_S(w) = d_S(v')$ . By the induction hypothesis we have  $y_1^*, y_2^*, \dots, y_t^* \in N^{dist(v, S)-1}[v] \cap N(S)$ . Therefore, we have  $d_S(u) \geq \max_{1 \leq i \leq t} d_S(y_i^*)$ . We also have  $d_S(w) \geq d_S(y'_i)$  for any  $i$  such that  $1 \leq i \leq t$  and  $d_S(u) = d_S(y_i^*)$ . Furthermore, by maximality of  $d_S(u)$  we must have  $u \in N(S)$ . In particular, there exists a vertex  $x$  such that  $x \in Proj(v, S)$  and  $u$  is on a shortest  $vx$ -path. Such a shortest path must contain  $y_i$ , for some  $i$  such that  $1 \leq i \leq t$ , thus implying by the induction hypothesis  $d_S(y_i^*) \geq d_S(u)$ . Then,  $d_S(y_i^*) = d_S(u)$ , and so,  $d_S(y'_i) = d_S(w)$ . As a result,

$$d_S(u) = \max_{1 \leq i \leq t} d_S(y_i^*) = d_S(v^*),$$

and in the same way

$$d_S(w) = \max\{d_S(y'_i) \mid 1 \leq i \leq t \text{ and } d_S(y_i^*) = d_S(v^*)\} = d_S(v'),$$

thus proving correctness of the algorithm for vertex  $v$ .

Finally, let us consider the two vertices  $v^*, v'$  that are mapped to some arbitrary vertex  $v$ . Let us assume that vertex  $v$  has a gate  $u$  in  $S$ . By the maximality of  $d_S(v^*)$  within  $N^{\text{dist}(v,S)-1}[v]$ ,  $d_S(v^*) \geq d_S(u) = |\text{Proj}(v, S)|$ . Since in addition, every vertex of  $N(v^*) \cap S$  is in  $\text{Proj}(v, S)$ , we obtain that  $\text{Proj}(v, S) \subseteq N(v^*)$ . Then,  $v^*$  is a gate of  $v$ . In the same way, let us assume that vertex  $v$  has a pseudo-gate in  $S$ . In particular, vertex  $v$  also has a gate in  $S$ . Then, vertex  $v^*$  is a gate of vertex  $v$  in  $S$ . Since vertex  $v^*$  is in  $N[v']$ ,  $\text{dist}(v, v') \leq \text{dist}(v, v^*) + 1 = \text{dist}(v, S)$ . Let  $w$  be some pseudo-gate of vertex  $v$  in  $S$ , and let  $u$  be some gate of vertex  $v$  in  $S$  such that  $u \in N[w]$ . We have that  $d_S(u) = d_S(v^*)$  because both vertices  $u, v^*$  are gates of vertex  $v$  in  $S$ . Then,  $d_S(v') \geq d_S(w)$  because  $(d_S(v^*), d_S(v'))$  is lexicographically maximized. Furthermore,  $N[w] \cap S \subseteq N^{\text{dist}(v,S)+1}[v]$ , and in the same way  $N[v'] \cap S \subseteq N^{\text{dist}(v,S)+1}[v]$ , because  $w, v' \in N^{\text{dist}(v,S)}[v]$ . Then,  $N[w] \cap S = N^{\text{dist}(v,S)+1}[v] \cap S$  because  $w$  is a pseudo-gate of  $v$  in  $S$ . It implies that  $N[v'] \cap S = N^{\text{dist}(v,S)+1}[v] \cap S$  because  $d_S(v') \geq d_S(w)$ . Therefore,  $v'$  is a pseudo-gate of vertex  $v$  in  $S$ .  $\square$

Let  $\mathbf{1}[\cdot]$  be the Kronecker symbol (equal to 1 if the predicate in brackets is true, and to 0 otherwise). By combining Lemmata 4 and 5, we are now ready to prove our main technical lemma.

**Lemma 7.** *Let  $G = (V, E)$  be a Helly graph, equipped with a nonnegative cost function  $c$ . For any fixed choice of a vertex  $v$  and a vertex subset  $A$ , we can compute the following three values for every neighbour  $u \in N(v)$ , in total  $\mathcal{O}(m)$  time:*

- $q_+(u, v, A) = \sum_{w \in A} c(w) \cdot \mathbf{1}[\text{dist}(u, w) > \text{dist}(v, w)];$
- $q_=(u, v, A) = \sum_{w \in A} c(w) \cdot \mathbf{1}[\text{dist}(u, w) = \text{dist}(v, w)];$
- $q_-(u, v, A) = \sum_{w \in A} c(w) \cdot \mathbf{1}[\text{dist}(u, w) < \text{dist}(v, w)].$

*Proof.* Let  $S = N[v]$ . By Lemma 4, every vertex  $w \notin S$  has a gate. Furthermore, by Lemma 6 we can compute a fixed gate  $g(w)$  for every vertex  $w \notin S$ , in total  $\mathcal{O}(m)$  time. For every vertex  $z \in N(S)$ , let  $\alpha(z) = \sum_{w \in A \setminus S} c(w) \cdot \mathbf{1}[g(w) = z]$ . Let  $u \in S$  be an arbitrary neighbour of vertex  $v$ . Clearly,  $u$  is the only vertex  $w \in S$  such that  $\text{dist}(u, w) < \text{dist}(v, w)$ . For every vertex  $w$  not in  $S$ , we claim that we have  $\text{dist}(u, w) < \text{dist}(v, w)$  if and only if  $g(w) \in N(u)$ . Indeed, let us first assume that  $g(w) \in N(u)$ . Since  $g(w)$  is a gate of  $w$  in  $N(v)$ , we have  $\text{dist}(w, g(w)) = \text{dist}(w, N(v)) - 1 = \text{dist}(w, v) - 2$ . As a result,  $\text{dist}(w, u) \leq \text{dist}(w, g(w)) + \text{dist}(g(w), u) \leq \text{dist}(w, v) - 1$ . Conversely, we now assume that  $\text{dist}(u, w) < \text{dist}(v, w)$ . Since  $u$  and  $v$  are adjacent,  $\text{dist}(w, u) = \text{dist}(w, v) - 1$ . It implies that  $u \in \text{Proj}(w, N(v))$ . As a result,  $u$  and  $g(w)$  are adjacent, thus proving our claim. Therefore, for every neighbour  $u \in N(v)$ , we have:

$$\begin{aligned}
q_-(u, v, A) &= \sum_{w \in A} c(w) \cdot \mathbf{1}[\text{dist}(u, w) < \text{dist}(v, w)] \\
&= \sum_{w \in A \cap S} c(w) \cdot \mathbf{1}[\text{dist}(u, w) < \text{dist}(v, w)] + \sum_{w \in A \setminus S} c(w) \cdot \mathbf{1}[\text{dist}(u, w) < \text{dist}(v, w)] \\
&= c(u) \cdot \mathbf{1}[u \in A] + \sum_{w \in A \setminus S} c(w) \cdot \mathbf{1}[g(w) \in N(u)] \\
&= c(u) \cdot \mathbf{1}[u \in A] + \sum_{z \in N(S) \cap N(u)} \sum_{w \in A \setminus S} c(w) \cdot \mathbf{1}[g(w) = z] \\
&= c(u) \cdot \mathbf{1}[u \in A] + \sum_{z \in N(S) \cap N(u)} \alpha(z).
\end{aligned}$$

Being given the values  $\alpha(z)$  for every vertex  $z \in N(S)$  (precomputed at the beginning of the algorithm), we can compute  $q_-(u, v, A)$  in  $\mathcal{O}(|N(u)|)$  time for every neighbour  $u \in N(v)$ .

In the same way, by Lemma 5, every vertex  $w \notin S$  has a pseudo-gate. By Lemma 6 we can also compute a fixed pseudo-gate  $pg(w)$  for every vertex  $w \notin S$ , in total  $\mathcal{O}(m)$  time. For every vertex  $z \in N[S]$ , let  $\beta(z) = \sum_{w \in A \setminus S} c(w) \cdot \mathbf{1}[pg(w) = z]$ . Let  $w$  be an arbitrary vertex and let  $u \in S$  be a neighbour of  $v$ . If  $w \in S$ , then we have  $dist(u, w) \leq dist(v, w)$  if and only if  $w \in N[u] \setminus \{v\}$ . From now on, we assume that  $w$  is not in  $S$ . We claim that we have  $dist(u, w) \leq dist(v, w)$  if and only if  $pg(w) \in N[u]$ . Indeed, let us first consider the case where  $pg(w) \in N[u]$ . We have  $dist(w, pg(w)) \leq dist(w, S) = dist(w, v) - 1$ . Therefore,  $dist(w, u) \leq dist(w, pg(w)) + 1 \leq dist(w, v)$ . Conversely, we now consider the case where  $dist(u, w) \leq dist(v, w)$ . We have  $dist(u, w) \leq dist(v, w) = r$ , with  $r = dist(w, S) + 1$ . In particular,  $u \in N^r[w] \cap S$ . It implies that  $u \in N[pg(w)]$ , or equivalently  $pg(w) \in N[u]$ , thus proving our claim. We obtain that for every neighbour  $u \in N(v)$ :

$$\begin{aligned}
q_-(u, v, A) + q_=(u, v, A) &= \sum_{w \in A} c(w) \cdot \mathbf{1}[dist(u, w) \leq dist(v, w)] \\
&= \sum_{u' \in S \cap A} c(u') \cdot \mathbf{1}[dist(u, u') \leq dist(v, u')] + \sum_{w \in A \setminus S} c(w) \cdot \mathbf{1}[dist(u, w) \leq dist(v, w)] \\
&= \sum_{u' \in N(v) \cap N[u] \cap A} c(u') + \sum_{w \in A \setminus S} c(w) \cdot \mathbf{1}[pg(w) \in N[u]] \\
&= \sum_{u' \in N(v) \cap N[u] \cap A} c(u') + \sum_{z \in N[S] \cap N[u]} \sum_{w \in A \setminus S} c(w) \cdot \mathbf{1}[pg(w) = z] \\
&= \sum_{u' \in N(v) \cap N[u] \cap A} c(u') + \sum_{z \in N[S] \cap N[u]} \beta(z).
\end{aligned}$$

For every neighbour  $u \in N(v)$ , being given  $q_-(u, v, A)$  and the values  $\beta(z)$  for every vertex  $z \in N[S]$ , we can compute  $q_=(u, v, A)$  in  $\mathcal{O}(|N(u)|)$  time.

Finally,  $q_+(u, v, A) = \sum_{w \in A} c(w) - q_-(u, v, A) - q_=(u, v, A)$ . □

## Main Results.

**Theorem 1.** *If  $G$  is a Helly graph then, for any nonnegative cost function  $c$ , w.h.p., we can compute a central vertex in  $\tilde{\mathcal{O}}(m\sqrt{n})$  time.*

*Proof.* Set  $p = n^{-\frac{1}{2}}$ , and let  $U(p)$  be a corresponding random subset. We observe that the expected cardinality of  $U(p)$  equals  $pn = \sqrt{n}$ . By Chernoff bounds,

$$\Pr[|U(p)| > 2\sqrt{n}] \leq e^{-\frac{\sqrt{n}}{3}}.$$

In particular, the subset  $U(p)$  has cardinality  $\tilde{\mathcal{O}}(\sqrt{n})$  w.h.p., and thus we assume from now on that it is indeed the case. We compute the  $c$ -eccentricity for every vertex of  $U(p)$ , that takes  $\mathcal{O}(m|U(p)|) = \tilde{\mathcal{O}}(m\sqrt{n})$  time if we run a BFS starting from every vertex of  $U(p)$ . Then, let  $u \in U(p)$  be of minimum  $c$ -eccentricity. At each step of the algorithm, we search for a neighbour  $v$  of the current vertex  $u$  such that  $e_c(v) < e_c(u)$ . If no such neighbour exists then,  $u$  is a local minimum for the eccentricity function, and so, by Lemma 1, this vertex  $u$  is central. Otherwise, we set  $u = v$  and then we continue the algorithm for at least one more step. All the vertices  $u$  considered during

the algorithm induce an eccentricity-monotone path. By Lemma 1, the eccentricity function of  $G$  is unimodal. Therefore, by Lemma 3 the total number of steps is upper bounded w.h.p. by  $\tilde{\mathcal{O}}(\sqrt{n})$ .

We can implement each step of this local search algorithm in  $\mathcal{O}(m)$  time, as follows. First we observe that for every vertex  $u$  and any neighbour  $v \in N(u)$ , for every vertex  $w$  we have  $|\text{dist}(v, w) - \text{dist}(u, w)| \leq 1$ . Let  $A = \{w \in V \mid c(w) \cdot \text{dist}(u, w) = e_c(u)\}$  and let  $B = \{w \in V \mid c(w) \cdot (\text{dist}(u, w) + 1) \geq e_c(u)\}$ . We have  $e_c(v) < e_c(u)$  if and only if the following two conditions hold: (i)  $q_+(v, u, A) = q_-(v, u, A) = 0$ , and (ii)  $q_+(v, u, B) = 0$ . Therefore, by applying Lemma 7 twice, we can decide in  $\mathcal{O}(m)$  time whether there exists a neighbour  $v \in N(u)$  such that  $e_c(v) < e_c(u)$ , and if so, compute such a neighbour within the same amount of time.  $\square$

**Theorem 2.** *If  $G$  is a Helly graph then, for any nonnegative cost function  $c$ , w.h.p., we can compute all the medians in  $\tilde{\mathcal{O}}(m\sqrt{n})$  time.*

*Proof.* First, we compute one median. Since by Lemma 2, the total  $c$ -distance function of a Helly graph is unimodal, we can do a local search algorithm, similar to the one presented in the proof of Theorem 1 but on a different unimodal function. However, we need to explain how we can implement each step of this algorithm in  $\mathcal{O}(m)$  time. For that, let  $u \in V$  be fixed and let  $v \in N(u)$  be an arbitrary neighbour. Again, recall that for every vertex  $w$  we have  $|\text{dist}(v, w) - \text{dist}(u, w)| \leq 1$ . Then, we have

$$\begin{aligned}
TD_c(u) - TD_c(v) &= \sum_{w \in V} c(w) \cdot \text{dist}(u, w) - \sum_{w \in V} c(w) \cdot \text{dist}(v, w) \\
&= \sum_{w \in V} c(w) \cdot (\text{dist}(u, w) - \text{dist}(v, w)) \\
&= \sum_{w \in V} c(w) \cdot 1 \cdot \mathbf{1}[\text{dist}(u, w) > \text{dist}(v, w)] \\
&\quad + \sum_{w \in V} c(w) \cdot 0 \cdot \mathbf{1}[\text{dist}(u, w) = \text{dist}(v, w)] \\
&\quad + \sum_{w \in V} c(w) \cdot (-1) \cdot \mathbf{1}[\text{dist}(u, w) < \text{dist}(v, w)] \\
&= \sum_{w \in V} c(w) \cdot \mathbf{1}[\text{dist}(u, w) > \text{dist}(v, w)] - \sum_{w \in V} c(w) \cdot \mathbf{1}[\text{dist}(u, w) < \text{dist}(v, w)] \\
&= q_-(v, u, V) - q_+(v, u, V).
\end{aligned}$$

Therefore, by applying Lemma 7 once, we can decide in  $\mathcal{O}(m)$  time whether there exists a neighbour  $v \in N(u)$  such that  $TD_c(v) < TD_c(u)$ , and if so, compute such a neighbour within the same amount of time. – In fact, we can even compute a vertex  $v$  that minimizes  $TD_c(v)$  within  $N(u)$ . However, it is not clear whether there is a way to exploit this for a faster algorithm. –

Finally, let  $u \in V$  be a median. Again by using Lemma 7, in  $\mathcal{O}(m)$  time we can compute all the neighbours  $v \in N(u)$  such that  $TD_c(u) = TD_c(v)$ . Since the subset of all medians of a Helly graph induces a complete subgraph (Lemma 2), we computed doing so all the medians.  $\square$

### 3 Radius computation within $k$ -Helly graphs

Recall that in [36], we proved that the radius of a Helly graph can be computed w.h.p. in  $\tilde{\mathcal{O}}(m\sqrt{n})$  time. In this last section, we generalize this result to the  $k$ -Helly graphs. The radius of a graph  $G$

is denoted in what follows by  $rad(G) = \min_v e(v)$ .

**Theorem 3.** *If  $G$  is a  $k$ -Helly graph, then there is an algorithm that runs in  $\tilde{O}(m\sqrt{kn})$  time w.h.p., and computes  $rad(G)$ .*

Theorem 3 follows from a more general result, that we state next. Recall that we call a graph  $(k, \alpha)$ -Helly if, for every family of  $k$ -wise intersecting balls  $N^{r_1}[v_1], N^{r_2}[v_2], \dots, N^{r_s}[v_s]$ , there exists a vertex  $x$  such that for every  $i$  with  $1 \leq i \leq s$ ,  $dist(x, v_i) \leq r_i + \alpha$ . Note that in particular, the  $(k, 0)$ -Helly graphs are exactly the  $k$ -Helly graphs. It is also known that the chordal graphs are  $(2, \mathcal{O}(1))$ -Helly, and more generally the  $k$ -hyperbolic graphs are  $(2, \mathcal{O}(k))$ -Helly [23].

**Theorem 4.** *If  $G$  is a  $(k, \alpha)$ -Helly graph then there is an algorithm that runs in  $\tilde{O}(m\sqrt{kn})$  time w.h.p., and computes some value  $r$  such that  $rad(G) - \alpha \leq r \leq rad(G)$  (hereafter called an additive  $+\alpha$ -approximation of  $rad(G)$ ).*

We observe that Theorem 3 is an easy corollary of Theorem 4. The remainder of this section is devoted to the proof of Theorem 4. For that, we need the following lemma, that is based on the same random sampling technique as for Lemma 3.

**Lemma 8** (Lemma 2 in [36]). *Let  $G = (V, E)$  be a graph, let  $r$  be a positive integer and let  $\varepsilon \in (0; 1)$ . There is an algorithm that computes a set  $D\langle G; r; \varepsilon \rangle$  such that the following two properties hold for every vertex  $v$ :*

- *if  $e(v) \leq r$  then  $v \in D\langle G; r; \varepsilon \rangle$ ;*
- *conversely, if  $v \in D\langle G; r; \varepsilon \rangle$  then w.h.p.  $|N^r[v]| \geq (1 - \varepsilon) \cdot n$ .*

Furthermore, w.h.p. the algorithm runs in  $\mathcal{O}(m \log n / \varepsilon)$  time.

Equipped with Lemma 8, we are now ready to prove the following decision version of Theorem 4:

**Lemma 9.** *Let  $G = (V, E)$  be a  $(k, \alpha)$ -Helly graph, and let  $r$  be a positive integer. There is an algorithm that runs in  $\tilde{O}(m\sqrt{kn})$  time w.h.p., and satisfies the following two properties:*

- *If the algorithm rejects, then  $rad(G) > r$ ;*
- *If the algorithm accepts, then  $rad(G) \leq r + \alpha$ .*

We prove next that by a classical dichotomic argument, Lemma 9 is equivalent to Theorem 4. Note that in order to prove Theorem 4, it suffices to prove that it is implied by Lemma 9. The converse is proved only for the sake of completeness.

We start proving that Theorem 4 implies Lemma 9. Let  $r$  be an arbitrary positive integer, and let  $r'$  be the output of Theorem 4. We can design a simple algorithm that accepts if and only if  $r \leq r' + \alpha$ . Indeed, recall that  $rad(G) - \alpha \leq r' \leq rad(G)$ . Therefore, if  $r \leq r' + \alpha$  then we have  $r \leq rad(G) + \alpha$ , and conversely if  $r > r' + \alpha$  then we have  $r > rad(G)$ .

We now present an algorithm in order to prove Theorem 4, assuming Lemma 9. More specifically, at any step  $i \geq 0$  of the algorithm we maintain a range  $[rm_i; ru_i]$  so that  $[rm_i; ru_i] \cap [rad(G) - \alpha; rad(G)] \neq \emptyset$  (initially,  $rm_0 = 1$  and  $ru_0 = n - 1$ ). While  $rm_i \neq ru_i$ , we pick  $r_i = \lfloor (rm_i + ru_i) / 2 \rfloor$  and we apply Lemma 9. There are two cases:

- If  $\text{rad}(G) > r_i$ , then we set  $rm_{i+1} = r_i + 1$  and  $ru_{i+1} = ru_i$ . We claim that in this situation, we still have  $[rm_{i+1}; ru_{i+1}] \cap [\text{rad}(G) - \alpha; \text{rad}(G)] \neq \emptyset$ . Indeed, if it were not the case, then since we assume that  $\text{rad}(G) \geq r_i + 1 = rm_{i+1}$ , we should have that  $\text{rad}(G) - \alpha > ru_{i+1} = ru_i$ , thus contradicting that we were having  $[rm_i; ru_i] \cap [\text{rad}(G) - \alpha; \text{rad}(G)] \neq \emptyset$  at the prior step of the algorithm.
- Otherwise,  $\text{rad}(G) \leq r_i + \alpha$ . In this situation, we set  $rm_{i+1} = rm_i$ ,  $ru_{i+1} = r_i$  and we claim that we still have  $[rm_{i+1}; ru_{i+1}] \cap [\text{rad}(G) - \alpha; \text{rad}(G)] \neq \emptyset$ . Indeed, if it were not the case, then since we assume that  $\text{rad}(G) \leq r_i + \alpha = ru_{i+1} + \alpha$ , or equivalently  $\text{rad}(G) - \alpha \leq ru_{i+1}$  we should have that  $\text{rad}(G) < rm_{i+1} = rm_i$ , thus contradicting that we were having  $[rm_i; ru_i] \cap [\text{rad}(G) - \alpha; \text{rad}(G)] \neq \emptyset$  at the prior step of the algorithm.

Overall, we can only call Lemma 9  $\mathcal{O}(\log n)$  times before the algorithm stops.

*Proof of Lemma 9.* We first describe the algorithm, before proving its correctness and then analyzing its running time. Let  $\varepsilon > 0$  to be fixed later in the proof. We proceed as follows:

- We compute a set  $C_0 = D\langle G; r; \varepsilon \rangle$  such as in Lemma 8. Such a set  $C_0$  can be computed w.h.p. in  $\mathcal{O}(m \log n / \varepsilon)$  time. Furthermore, if  $C_0 = \emptyset$ , then  $\text{rad}(G) > r$ , and we reject.
- Then, for  $i = 1 \dots, k$ , we select an arbitrary vertex  $v_{i-1} \in C_{i-1}$  and we compute  $N^r[v_{i-1}]$ . It takes linear time. If  $e(v_{i-1}) \leq r$ , then  $\text{rad}(G) \leq r$ , and we accept. Otherwise, let  $S_i = V \setminus N^r[v_{i-1}]$ . We compute  $C_i = C_{i-1} \cap \bigcap \{N^r[s_i] \mid s_i \in S_i\}$ . It can be done in  $\mathcal{O}(|S_i|m)$  time if we run a BFS on every vertex of  $S_i$ . By construction,  $v_{i-1} \in C_0$ . Therefore by Lemma 8,  $|S_i| \leq \varepsilon n$  w.h.p., and so, w.h.p. the computation of  $C_i$  takes  $\mathcal{O}(mn\varepsilon)$  time. Furthermore, if  $C_i = \emptyset$ , then we reject.
- Finally, if  $C_k \neq \emptyset$ , then we accept.

In order to prove correctness of this algorithm, let us first discuss the case where it ends during the for loop. If we select some vertex  $v_{i-1}$  such that  $e(v_{i-1}) \leq r$ , then clearly  $\text{rad}(G) \leq r$ . Therefore, we only need to discuss the subcase where  $C_i = \emptyset$  for some  $i$ . For every  $j$  such that  $0 \leq j \leq i$ , we claim that  $C_j$  must contain all vertices of eccentricity at most  $r$  (if any). We prove it by induction. For  $j = 0$ , the result follows from Lemma 8. Thus, from now on, we assume that  $j > 0$  and that the claim holds for  $j - 1$ . Every vertex of eccentricity at most  $r$  must be contained in  $\bigcap \{N^r[s_j] \mid s_j \in S_j\}$ , and therefore by the induction hypothesis it must be contained in  $C_j$ . As a result, we proved that all vertices of eccentricity at most  $r$  are contained in  $C_i$ . Since we here assume that  $C_i = \emptyset$ , we obtain that  $\text{rad}(G) > r$ .

From this point on, we focus on the last step of the algorithm. Specifically, we claim that if  $C_k \neq \emptyset$ , then  $\text{rad}(G) \leq r + \alpha$ . Suppose by contradiction that  $\text{rad}(G) > r + \alpha$ . Since we assume  $G$  to be  $(k, \alpha)$ -Helly, there exists a  $k$ -subset  $A$  such that  $\bigcap \{N^r[a] \mid a \in A\} = \emptyset$ . Clearly, we cannot have  $A \subseteq N^r[v_{i-1}]$ , for any  $i$  such that  $1 \leq i \leq k$ . Furthermore by construction the sets  $S_1, S_2, \dots, S_k$  are pairwise disjoint. Therefore, for every  $i$  such that  $1 \leq i \leq k$ ,  $|A \cap S_i| = 1$ . However, by construction we also have  $C_k = C_0 \cap \bigcap \{N^k[s] \mid s \in \bigcup_{i=1}^k S_i\}$ . Therefore,  $\emptyset \neq C_k \subseteq \bigcap \{N^r[a] \mid a \in A\} = \emptyset$ , a contradiction.

Overall, the total running time is in  $\mathcal{O}(m \log n / \varepsilon + kmn\varepsilon)$ . This is optimized when  $\varepsilon = \Theta(\sqrt{\log n / (kn)})$ , and then the running time is in  $\mathcal{O}(m\sqrt{kn \log n})$ .  $\square$

**Acknowledgements.** We thank the anonymous reviewers for their valuable feedback. This work was supported by a grant of the Ministry of Research, Innovation and Digitalization, CCCDI - UEFISCDI, project number PN-III-P2-2.1-PED-2021-2142, within PNCDI III.

## References

- [1] A. Abboud, V. Vassilevska Williams, and J. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proceedings of the twenty-seventh annual Symposium on Discrete Algorithms (SODA)*, pages 377–391. SIAM, 2016.
- [2] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM Journal on Computing*, 28(4):1167–1181, 1999.
- [3] N. Amenta. Helly-type theorems and generalized linear programming. *Discrete & Computational Geometry*, 12(3):241–261, 1994.
- [4] H. Bandelt and V. Chepoi. Metric graph theory and geometry: a survey. *Contemporary Mathematics*, 453:49–86, 2008.
- [5] H.-J. Bandelt and V. Chepoi. Graphs with connected medians. *SIAM Journal on Discrete Mathematics*, 15(2):268–282, 2002.
- [6] H.-J. Bandelt and E. Pesch. Dismantling absolute retracts of reflexive graphs. *European Journal of Combinatorics*, 10(3):211–220, 1989.
- [7] H.-J. Bandelt and E. Pesch. A Radon theorem for Helly graphs. *Archiv der Mathematik*, 52(1):95–98, 1989.
- [8] H.-J. Bandelt and E. Prisner. Clique graphs and Helly graphs. *Journal of Combinatorial Theory, Series B*, 51(1):34–45, 1991.
- [9] L. Bénéteau, J. Chalopin, V. Chepoi, and Y. Vaxès. Medians in median graphs and their cube complexes in linear time. *Journal of Computer and System Sciences*, 126:80–105, 2022.
- [10] C. Berge. *Graphs and hypergraphs*. North-Holland Publishing Company, 1973.
- [11] J. A. Bondy and U. S. R. Murty. *Graph theory*. Springer, 2008.
- [12] A. Brandstädt, V. Chepoi, and F. Dragan. The algorithmic use of hypertree structure and maximum neighbourhood orderings. *Discrete Applied Mathematics*, 82(1-3):43–77, 1998.
- [13] A. Brandstädt, F. Dragan, V. Chepoi, and V. Voloshin. Dually chordal graphs. *SIAM Journal on Discrete Mathematics*, 11(3):437–455, 1998.
- [14] K. Bringmann, T. Husfeldt, and M. Magnusson. Multivariate analysis of orthogonal range searching and graph distances. *Algorithmica*, 82(8):2292–2315, 2020.
- [15] S. Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. *ACM Transactions on Algorithms*, 15(2):21, 2018.

- [16] S. Cabello and C. Knauer. Algorithms for graphs of bounded treewidth via orthogonal range searching. *Computational Geometry*, 42(9):815–824, 2009.
- [17] J. Chalopin, V. Chepoi, A. Genevois, H. Hirai, and D. Osajda. Helly groups. Technical Report arXiv:2002.06895, arXiv, 2020.
- [18] G. Chang and G. Nemhauser. The  $k$ -domination and  $k$ -stability problems on sun-free chordal graphs. *SIAM Journal on Algebraic Discrete Methods*, 5(3):332–345, 1984.
- [19] M. Chastand, F. Laviolette, and N. Polat. On constructible graphs, infinite bridged graphs and weakly cop-win graphs. *Discrete Mathematics*, 224(1-3):61–78, 2000.
- [20] V. Chepoi and F. Dragan. A linear-time algorithm for finding a central vertex of a chordal graph. In *Proceedings of the second annual European Symposium on Algorithms (ESA)*, pages 159–170. Springer, 1994.
- [21] V. Chepoi and F. Dragan. Finding a central vertex in an HHD-free graph. *Discrete Applied Mathematics*, 131(1):93–111, 2003.
- [22] V. Chepoi, F. Dragan, and Y. Vaxès. Center and diameter problems in plane triangulations and quadrangulations. In *Proceedings of the thirteenth annual Symposium on Discrete Algorithms (SODA)*, pages 346–355. ACM-SIAM, 2002.
- [23] V. Chepoi and B. Estellon. Packing and covering  $\delta$ -hyperbolic spaces by balls. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 59–73. Springer, 2007.
- [24] K. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *Journal of the ACM (JACM)*, 42(2):488–499, 1995.
- [25] D. Corneil, F. Dragan, M. Habib, and C. Paul. Diameter determination on restricted graph families. *Discrete Applied Mathematics*, 113(2-3):143–166, 2001.
- [26] D. Coudert, G. Ducoffe, and A. Popa. Fully polynomial FPT algorithms for some classes of bounded clique-width graphs. *ACM Transactions on Algorithms*, 15(3), 2019.
- [27] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- [28] F. Dragan. *Centers of graphs and the Helly property*. PhD thesis, Moldova State University, 1989.
- [29] F. Dragan. Domination in quadrangle-free Helly graphs. *Cybernetics and Systems Analysis*, 29(6):822–829, 1993.
- [30] F. Dragan and A. Brandstädt.  $r$ -dominating cliques in graphs with hypertree structure. *Discrete Mathematics*, 162(1-3):93–108, 1996.
- [31] F. Dragan, G. Ducoffe, and H. Guarnera. Fast deterministic algorithms for computing all eccentricities in (hyperbolic) Helly graphs. In *Proceedings of the seventeenth Workshop on Algorithms and Data Structures (WADS)*, pages 300–314. Springer, 2021.



- [32] F. Dragan and H. Guarnera. Helly-gap of a graph and vertex eccentricities. *Theoretical Computer Science*, 867:68–84, 2021.
- [33] F. Dragan and F. Nicolai. LexBFS-orderings of distance-hereditary graphs with application to the diametral pair problem. *Discrete Applied Mathematics*, 98(3):191–207, 2000.
- [34] F. Dragan, F. Nicolai, and A. Brandstädt. LexBFS-orderings and powers of graphs. In *Proceedings of the twenty-third International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 166–180. Springer, 1997.
- [35] A. Dress. Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups: a note on combinatorial properties of metric spaces. *Advances in Mathematics*, 53(3):321–402, 1984.
- [36] G. Ducoffe and F. Dragan. A story of diameter, radius, and (almost) helly property. *Networks*, 77(3):435–453, 2021.
- [37] G. Ducoffe, M. Habib, and L. Viennot. Diameter computation on  $H$ -minor free graphs and graphs of bounded (distance) VC-dimension. In *Proceedings of the thirty-first annual Symposium on Discrete Algorithms (SODA)*, pages 1905–1922. SIAM, 2020.
- [38] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3-4):275–291, 2000.
- [39] A. Farley and A. Proskurowski. Computation of the center and diameter of outerplanar graphs. *Discrete Applied Mathematics*, 2(3):185–191, 1980.
- [40] P. Gawrychowski, H. Kaplan, S. Mozes, M. Sharir, and O. Weimann. Voronoi diagrams on planar graphs, and computing the diameter in deterministic  $\tilde{O}(n^{5/3})$  time. In *Proceedings of the twenty-ninth annual Symposium on Discrete Algorithms (SODA)*, pages 495–514. SIAM, 2018.
- [41] A. Goldman. Optimal center location in simple networks. *Transportation science*, 5(2):212–221, 1971.
- [42] M. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57. Elsevier, 2004.
- [43] J. Isbell. Six theorems about injective metric spaces. *Commentarii Mathematici Helvetici*, 39(1):65–76, 1964.
- [44] R. Laskar and D. Shier. On chordal graphs. *Congressus Numerantium*, 29:579–588, 1980.
- [45] M. Lin and J. Szwarcfiter. Faster recognition of clique-Helly and hereditary clique-Helly graphs. *Information Processing Letters*, 103(1):40–43, 2007.
- [46] J. Matousek. Bounded VC-dimension implies a fractional Helly theorem. *Discrete & Computational Geometry*, 31(2):251–255, 2004.
- [47] S. Olariu. A simple linear-time algorithm for computing the center of an interval graph. *International Journal of Computer Mathematics*, 34(3-4):121–128, 1990.

- [48] N. Polat. Convexity and fixed-point properties in Helly graphs. *Discrete Mathematics*, 229(1-3):197–211, 2001.
- [49] N. Polat. On constructible graphs, locally Helly graphs, and convexity. *Journal of Graph Theory*, 43(4):280–298, 2003.
- [50] L. Roditty and V. Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings of the forty-fifth Symposium on Theory of Computing (STOC)*, pages 515–524. ACM, 2013.
- [51] R. Seidel. On the all-pairs-shortest-path problem. In *Proceedings of the twenty-fourth Symposium on Theory of Computing (STOC)*, pages 745–749. ACM, 1992.
- [52] L. Viennot, M. Habib, and G. Ducoffe. Fast Diameter Computation within Split Graphs. *Discrete Mathematics & Theoretical Computer Science*, 23, 2021.