The Complexity of Diameter on H-free graphs*

Jelle J. Oostveen[1][0009−0009−4419−3143], Daniël Paulusma[2][0000−0001−5945−9287], and Erik Jan van Leeuwen[1][0000−0001−5240−7257]

[1] Dept. Information and Computing Sciences, Utrecht University, The Netherlands
j.j.oostveen@uu.nl, e.j.vanleeuwen@uu.nl
[2] Dept. Computer Science, Durham University, United Kingdom
daniel.paulusma@durham.ac.uk

Abstract. The intensively studied Diameter problem is to find the diameter of a given connected graph. We investigate, for the first time in a structured manner, the complexity of Diameter for H-free graphs, that is, graphs that do not contain a fixed graph H as an induced subgraph. We first show that if H is not a linear forest with small components, then Diameter cannot be solved in subquadratic time for H-free graphs under SETH. For some small linear forests, we do show linear-time algorithms for solving Diameter. For other linear forests H, we make progress towards linear-time algorithms by considering specific diameter values. If H is a linear forest, the maximum value of the diameter of any graph in a connected H-free graph class is some constant $d_{\max}$ dependent only on H. We give linear-time algorithms for deciding if a connected H-free graph has diameter $d_{\max}$, for several linear forests H. In contrast, for one such linear forest H, Diameter cannot be solved in subquadratic time for H-free graphs under SETH. Moreover, we even show that, for several other linear forests H, one cannot decide in subquadratic time if a connected H-free graph has diameter $d_{\max}$ under SETH.

1 Introduction

The Diameter problem asks to find the diameter of an undirected, unweighted graph $G = (V, E)$, that is, the longest of the shortest paths between all pairs of nodes; formally, $\mathrm{diam}(G) = \max_{u,v \in V} d(u, v)$. We shall denote $|V| = n$ and $|E| = m$. A trivial algorithm executes a Breadth First Search (BFS) from every node in the graph, and has a running time of $O(nm)$. The best known matrix multiplication-based algorithms achieve a running time of $\tilde{O}(n^\omega)$ [16,43,45] to find the diameter of a graph, where $\tilde{O}$ hides logarithmic factors and ω is the matrix multiplication constant, with current known value $\omega < 2.371866$ [26]. A search for improvement led to a hardness result: under the Strong Exponential Time Hypothesis (SETH), one cannot decide between diameter 2 or 3 on (sparse) split graphs in $O(n^{2-\epsilon})$ time, for any $\epsilon > 0$ [42]. SETH is a hypothesis that states that Satisfiability cannot be solved in $2^{(1-\epsilon)n}$ time, for any $\epsilon > 0$, where n is the number of variables [37,38]. On other simple graph classes like

* J.J. Oostveen is supported by the NWO grant OCENW.KLEIN.114 (PACAN).

constant degree graphs, truly subquadratic time algorithms for DIAMETER are also ruled out under SETH [34]. Directed versions of the DIAMETER problem admit similar barriers under SETH [4]. No clear bound is known for DIAMETER on dense graphs, and no such lower bound can be derived when $\omega = 2$, but we do know that there is a subcubic equivalence between DIAMETER and computing the reach centrality of a graph, that is, a truly subcubic algorithm for one implies such an algorithm for the other and vice versa [1].

Given that the hardness results are based on long-standing conjectures, it is natural to approach diameter computation and other similar problems on restricted graph classes. Related literature also concerns computation of eccentricities, as computing the diameter of a graph is equivalent to computing the largest eccentricity over all vertices. A conceptually simple algorithm called LexBFS can solve DIAMETER in $O(n+m)$ time for distance-hereditary chordal graphs and interval graphs [25]. Distance-hereditary graphs have been studied separately, and admit linear-time algorithms of all eccentricities [19,22,24]. Interval graphs admit computation of the eccentricity of the center of the graph in linear time, next to linear-time diameter computation [40]. Subquadratic algorithms for DIAMETER and computing eccentricities have been studied for more graph classes, including asteroidal triple (AT)-free graphs [28], directed path graphs [13], strongly chordal graphs [17], dually chordal graphs [6,18], Helly graphs and graphs of bounded Helly number [21,30,31], α_i-metric graphs [20], retracts [27], δ-hyperbolic graphs [11,12,22,23], planar graphs [2,8,36], and outerplanar graphs [35]. DIAMETER was also studied from the parameterized perspective, see e.g. [4,7,15,29,32], and a large body of work exists on approximation algorithms, see e.g. [4,5,10,13,14,20,42,44].

For graph classes with forbidden patterns, Ducoffe et al. [32] show subquadratic-time algorithms of DIAMETER for $\mathcal{H}$-minor-free graphs, where the precise exponent depends on $\mathcal{H}$, improved upon by Duraj et al. [33]. Johnson et al. [39] showed for $\mathcal{H}$-subgraph-free graphs that there is a dichotomy for DIAMETER between almost-linear time solvability and quadratic-time conditional lower bounds depending on the family $\mathcal{H}$. Also note that many of the studied graph classes listed above can be characterized as $\mathcal{H}$-free graphs for a family of graphs $\mathcal{H}$, but for each one, $\mathcal{H}$ has size 2 or larger. As far as we are aware, a structured study into forbidden (monogenic) induced patterns is absent in the literature.

Our Contributions. We initiate a structured study into diameter computation on H-free graphs, where H is a single graph. Recall that a graph is H-free if it does not contain H as an induced subgraph. The question we consider is:

For which H-free graph classes $\mathcal{G}$ can the diameter of an n-vertex graph $G \in \mathcal{G}$ be computed in time $O(n^{2-\epsilon})$?

Our first result analyses existing lower bounds to find hardness for H-free graph classes. Here, P_t denotes the path on t vertices, and sP_t denotes the disjoint union of s copies of a P_t. For graphs G, H let $G + H$ denote their disjoint union. Recall that a *linear forest* is a disjoint union of one or more paths.

Theorem 1 (♠[3]). *Let H be a graph that contains an induced $2P_2$ or is not a linear forest. Under SETH,* Diameter *on H-free graphs cannot be solved in $O(n^{2-\epsilon})$ time for any $\epsilon > 0$.*

Theorem 1 shows the most prominent gap to be for graph classes which exclude a small linear forest. Ducoffe [28] proved a hardness result for Diameter on AT-free graphs that holds under the hypothesis that the currently asymptotically fastest (combinatorial) algorithms for finding simplicial vertices (vertices with a complete neighbourhood) are optimal, which we refer to as the *Simplicial Vertex Hypothesis*. Because the vertex set of the graph in Ducoffe's hardness construction can be partitioned into four cliques, the result of [28] can be formulated as follows.

Theorem 2 ([28]). *For any fixed $k \geq 5$, under the Simplicial Vertex Hypothesis, there does not exist a combinatorial algorithm for* Diameter *on kP_1-free graphs that runs in $O(m^{3/2-\epsilon})$ time for any $\epsilon > 0$.*

To complement the hardness results, we show linear-time algorithms for several classes of H-free graphs for which H is a small linear forest.

Theorem 3 (♠). *Let H be a graph. If H is an induced subgraph of $P_2 + 2P_1$, $P_3 + P_1$, or P_4, then* Diameter *on H-free graphs can be solved in $O(n + m)$ time.*

We achieve Theorem 3 by careful structural analysis of the graph class and then show that a constant number of Breadth First Searches suffice algorithmically. Note that a running time of $O(n + m)$ clearly beats the naive algorithm of $O(nm)$ time and the matrix multiplication algorithms of $\tilde{O}(n^{\omega})$ time, but also rules out any quadratic lower bound in n, as the classes of graphs contain abitrarily large families of sparse graphs, e.g. stars.

Combining Theorems 1, 2 and 3, the only open cases for the complexity of Diameter on H-free graphs are $H = 4P_1$, $H = P_2 + 3P_1$, $H = P_3 + 2P_1$, $H = P_4 + 2P_1$, and $H = P_4 + P_1$. The smallest graph H that is an open case is that of $H = 4P_1$. As a 'hardness' result for $4P_1$-free graphs, one could try to take the split graph construction of Roditty and Williams [42], and add edges to make the graph consist of three cliques (as in [13]). Conceptually, this would seem to work: the diameter distinction is still 2 or 3 and translates to a SAT positive or negative answer. However, this approach fails due to the quantity of edges one adds to the graph. The lower bound shows that no $O(n^{2-\epsilon})$ time algorithm may exist for this new instance, which is now a relatively empty lower bound: the graph has a quadratic number of edges, so this lower bound does not even rule out an $O(n + m)$ time algorithm. The density of graphs matters in relation to lower bounds, and seems to provide a barrier to finding a lower bound that rules out a linear-time algorithm.

However, if we adopt the perspective from the other side, a linear-time algorithm for $4P_1$-free graphs would still be surprising. Indeed, such an algorithm

[3] Proofs of results marked with ♠ are deferred to the full version, see [41].

that can decide between diameter 2 or 3 on the three-clique instance described earlier implies an algorithm for ORTHOGONAL VECTORS in time $O(n^2 + d^2)$, where d is the dimension of the vectors and n the size of the vector sets (♠). Although lower bounds do not rule out this possibility, such a result would be highly non-trivial, as the best known algorithms for ORTHOGONAL VECTORS do not achieve this running time for all d [3,9]. Any linear-time algorithm would even beat the best known matrix-multiplication algorithms of $\tilde{O}(n^\omega)$ time, even if $\omega = 2$. It thus seems we are at an impasse to find or exclude a linear-time algorithm for computing the diameter of $4P_1$-free graphs.

However, as it turns out, we *can* decide in linear time whether the diameter of a $4P_1$-free graph is exactly 5. Our approach avoids the above barriers by focusing on specific diameter values instead of deciding on the diameter of a graph completely.

In general, for a graph class $\mathcal{G}$, we call $d_{\max}(\mathcal{G})$ the maximum diameter that any graph in $\mathcal{G}$ can have; formally $d_{\max}(\mathcal{G}) = \sup_{G \in \mathcal{G}} \operatorname{diam}(G)$. We omit $\mathcal{G}$ when it is clear from context. In particular, for $4P_1$-free graphs, $d_{\max}$ is equal to 5. We define the $d_{\max}(\mathcal{G})$-DIAMETER problem as deciding for a graph $G \in \mathcal{G}$ whether it holds that $\operatorname{diam}(G) = d_{\max}(\mathcal{G})$. The research question we investigate is:

For which H-free graph classes $\mathcal{G}$ can we solve $d_{\max}(\mathcal{G})$-DIAMETER in linear time?

For some classes $\mathcal{G}$, it is easy to see $d_{\max}(\mathcal{G})$ is bounded. For instance, the class of cliques has $d_{\max} = 1$. Any graph class that contains paths of arbitrary length has $d_{\max} = \infty$. For deciding whether the diameter of a graph is equal to $d_{\max}$, only classes with bounded $d_{\max}$ value are interesting to consider. It turns out that for classes of connected H-free graphs, $d_{\max}$ is bounded exactly when H is a linear forest (♠).

Our contributions with respect to the $d_{\max}(\mathcal{G})$-DIAMETER problem are twofold. Firstly, we find several examples of H-free classes $\mathcal{G}$ where H is a linear forest of more than one path where we solve $d_{\max}(\mathcal{G})$-DIAMETER in linear time. Note that $d_{\max}$ can differ vastly for classes where H is a linear forest, depending on H. Here, $H \subseteq_i H'$ denotes that H is an induced subgraph of H'.

Theorem 4 (♠). *Let $H = 2P_2 + P_1$ or $H \subseteq_i P_2 + 3P_1$, $P_3 + 2P_1$, or $P_4 + P_1$, and let $\mathcal{G}$ be the class of H-free graphs. Then $d_{\max}(\mathcal{G})$-DIAMETER can be solved in $O(n + m)$ time.*

Note that in particular, Theorem 4 shows that we can decide whether $\operatorname{diam}(G) = d_{\max}(\mathcal{G})$ for all previously stated open cases for DIAMETER computation on H-free graphs, except for the case of $H = P_4 + 2P_1$.

Secondly, we extend known hardness constructions to hold for the $d_{\max}(\mathcal{G})$-DIAMETER problem for certain H-free graph classes $\mathcal{G}$. Note that one needs different hardness proofs for different H, H', even if $H \subseteq_i H'$, because $d_{\max}$ can differ for both classes.

Theorem 5 (♠). *Let $H = 2P_2$ or $H = P_t$ for some odd $t \geq 5$, and let $\mathcal{G}$ be the class of H-free graphs. Under SETH, it is not possible to solve $d_{\max}(\mathcal{G})$-DIAMETER in time $O(n^{2-\epsilon})$ for any $\epsilon > 0$.*

Theorems 4 and 5 together cover almost all cases where $d_{\max} \leq 4$: $H = 2P_2$ is hard by Theorem 5, $H = P_3 + P_2$ is open, and all other cases with $d_{\max} \leq 4$ are linear-time solvable by Theorem 4. Theorem 4 also gives linear-time algorithms for some cases where $d_{\max} > 4$; $H = P_3 + 2P_1$ and $H = 2P_2 + P_1$ have $d_{\max} = 5$, and $H = P_2 + 3P_1$ has $d_{\max} = 6$. It appears that, algorithmically, the presence of a P_1 in H helps out in structural analysis, which may explain the inability to attain a result for $H = P_3 + P_2$. We further discuss particular cases and possible generalizations of our theorems in the conclusion.

Our algorithmic results are attained through careful analysis of the structure of the graph with respect to the forbidden pattern. This limits the ways in which a shortest path that realizes the diameter can appear in the graph. However, even for small patterns H, such analysis quickly becomes highly technical.

In this extended abstract, we give several sketches of our algorithmic technical proofs. We sketch a linear-time algorithm for Diameter on $(P_2 + 2P_1)$-free graphs, which also contributes to most of the linear-time algorithm for Diameter on $(P_3 + P_1)$-free graphs (♠). These two results together prove Theorem 3. We also sketch a part of the proof that we can solve $d_{\max}(\mathcal{G})$-Diameter on $(P_4 + P_1)$-free graphs in linear time. This result is the most technical result in our work, and contributes to Theorem 4.

Preliminaries. Graphs are denoted $G = (V, E)$ and are connected, undirected, and unweighted. For any $v \in V$, denote $N(v)$ as the neighbourhood of v, and $N[v] = N(v) \cup \{v\}$. For a set of vertices $S \subseteq V$, let $G[S]$ denote the induced subgraph on the vertices of S. A vertex $v \in V$ is *complete* to a set $S \subseteq V$ when $S \subseteq N(v)$, and *anti-complete* to a set S when $S \cap N(v) = \emptyset$. A set is $A \subseteq V$ is complete to a set $S \subseteq V$ when every vertex in A is complete to S, and A is anti-complete to S when every vertex in A is anti-complete to S. For vertices $v_1, \ldots, v_k \in V$, $\langle v_1, \ldots, v_k \rangle$ denotes an induced path from v_1 to v_k.

Two vertices $u, v \in V$ are *twins* when $N[u] = N[v]$, also called *true twins.* Two vertices $u, v \in V$ are *false twins* when $N(u) = N(v)$. We shall always be explicit when we talk about false twins; in general, twins will refer to true twins. A *twin class* is a set of vertices all of which are pairwise twins.

Removing twins can be done in linear time and keeps most distance properties in a graph (see e.g. Coudert et al. [15]).

Proposition 1. *Given a graph $G = (V, E)$, in $O(n + m)$ time we can detect true twins and remove all-but-one vertex from each twin class resulting in a graph $G' = (V', E')$ with $V' \subseteq V$, $E' \subseteq E$. The following hold:*

(i) the distance between two vertices u, v in G' is equal to the distance between u, v in G,
(ii) the diameter of G' is equal to the diameter of G, unless G is a clique,
(iii) if G is H-free for some graph H then G' is also H-free.

There are similar results with respect to false twins. The following was proved by Ducoffe [29] (formulated in terms of modules):

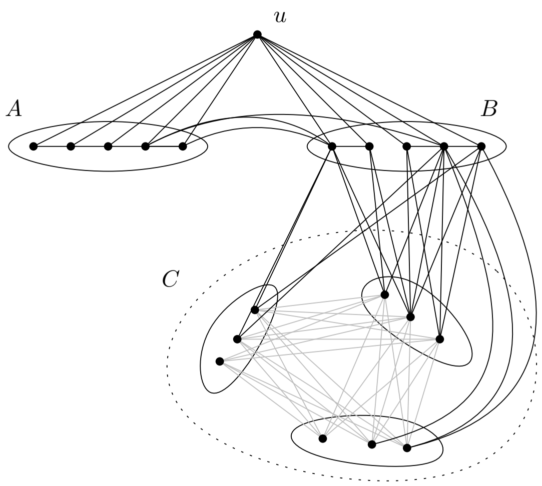

Fig. 1. A sketch of a $(P_2 + 2P_1)$-free graph as seen from some vertex u.

Corollary 1. *Given a graph G and a vertex set $B \subseteq V(G)$, we can partition B into classes of false twins with respect to their neighbourhoods towards $V(G) \setminus B$ in time $O(n + m)$.*

2 Algorithmic Results

2.1 $(P_2 + 2P_1)$-free graphs

We first prove that the diameter of a $(P_2 + 2P_1)$-free graph can be computed in linear time. The statement of the theorem is slightly stronger however, as we use this algorithm as a subroutine in another proof (♠).

Theorem 6 (♠). *Given a graph G, there is an algorithm that in $O(n + m)$ time either (a) correctly decides that G is not $(P_2 + 2P_1)$-free; or (b) outputs a shortest path, which is diametral if G is $(P_2 + 2P_1)$-free.*

Proof (Sketch). Let $G = (V, E)$ be a graph. The diameter of any $(P_2 + 2P_1)$-free graph is at most 4. If the diameter of G is 1, then the graph is a clique, which we can check in $O(n + m)$ time, and return any arbitrary pair of vertices, or a single vertex if $|V| = 1$.

Remove twins from the graph in $O(n + m)$ time. By Proposition 1, distances and the diameter are not affected, and the graph is H-free if it was H-free for some graph H. By abuse of notation, we still call this graph $G = (V, E)$.

Let u be a vertex in G with lowest degree, which can be found in $O(n + m)$ time. Now execute a BFS from u. We distinguish the structure of the graph as seen from u; see Figure 1. Let $C = V \setminus N[u]$ and let $A \subseteq N(u)$ be the subset of vertices of $N(u)$ with no edges to C. Let $B = N(u) \setminus A$. Note that A, B, C can be identified by the BFS from u. If $C = \emptyset$, then the diameter of G is at most 2, and we are done. Hence, $C \neq \emptyset$ and $B \neq \emptyset$. We note that $A = \emptyset$, which can be seen by the following. For any $a \in A$ it holds that $\deg(a) \leq \deg(u)$ by definition of A. u was picked to be a vertex of lowest degree in G, so for any $a \in A$ we have $\deg(a) = \deg(u)$. But as $N(a) \subseteq \{u\} \cup A \cup B$ for all $a \in A$ and $N(u) = A \cup B$, it follows that every $a \in A$ is a twin of u. But then $A = \emptyset$ as we removed twins.

If there is a vertex at distance 5 or more from u, then return that the graph is not $(P_2 + 2P_1)$-free. If there is a vertex at distance 4 from u, then return this shortest path; it is diametral if G is $(P_2 + 2P_1)$-free. Both of these cases are identified by the BFS from u. Now observe that any shortest path of length 3 or 4 must have at least one endpoint in C, as the distances between vertices in $A \cup B \cup \{u\}$ are at most 2 by u.

We prove properties of G under the assumption that it is $(P_2 + 2P_1)$-free.

Claim 1 (♠). *If G is $(P_2 + 2P_1)$-free, then (a) $G[C]$ is a complete r-partite graph for some $r \geq 1$; and (b) every $b \in B$ has at most one non-neighbour in every part of $G[C]$.*

Claim 2 (♠). *In $O(n + m)$ time, we can decide whether $G[C]$ is a complete r-partite graph for some $r \geq 1$ and, if so, return its parts.*

Claim 3 (♠). *In $O(n + m)$ time, either (a) we find a length-4 shortest path with both endpoints in C; (b) we find a length-3 shortest path with both endpoints in C and conclude no such length-4 shortest path exists in G; (c) we conclude no length-3 or length-4 shortest path with both endpoints in C exists in G; or (d) we conclude G is not $(P_2 + 2P_1)$-free.*

Run the algorithm of Claim 3. If it returns option (d), then output that G is not $(P_2 + 2P_1)$-free. If it returns option (a), then output the length-4 shortest path the algorithm gives; it is diametral if G is $(P_2 + 2P_1)$-free. In both other cases, we argue no length-4 diametral path can exist if G is $(P_2 + 2P_1)$-free. If G is $(P_2 + 2P_1)$-free, distances from vertices in B to vertices in C are at most 3, because $G[C]$ is r-partite and every $b \in B$ has at most one non-adjacent vertex per part of $G[C]$ by Claim 1. We already found a length-4 shortest path with u as an endpoint, if it exists. But then any length-4 shortest path has both endpoints in C, if it exists, as we already knew at least one endpoint was in C. In both options (b) and (c) we can conclude that no length-4 shortest path with both endpoints in C exists.

We continue as follows. If u has a vertex at distance 3, then we would already know this by the BFS from u. Otherwise, every $c \in C$ is at distance 2 from u. If the algorithm of Claim 3 returned option (b), we can output a shortest path of length 3 with both endpoints in C; it is diametral if G is $(P_2 + 2P_1)$-free. If the algorithm of Claim 3 returned option (c), no length-3 shortest path with both endpoints in C exists in G. Hence, the only remaining case for a length-3 shortest path is that there is a vertex in B with distance 3 to a vertex in C.

Claim 4 (♠). *If G is $(P_2 + 2P_1)$-free and a length-3 shortest path exists from some $b \in B$ to some $c \in C$, then (a) $G[C]$ has exactly one part with more than* *one vertex, and (b) all vertices in B with a vertex in C at distance 3 are adjacent to only that multi-vertex part and have exactly one non-neighbour in that part.*

By Claim 4, we can look for $G[C]$ to have simple structure. In particular, only one part may have multiple vertices. By Claim 2, we can detect if $G[C]$ is complete r-partite in $O(n+m)$ time, and, given the parts, check whether only one part has multiple vertices in $O(n+m)$ time. If this is not the case, then there is no length-3 shortest path from a vertex in B to a vertex in C by Claim 4, if G is (P_2+2P_1)-free, and we may return a length-2 shortest path with as witness some non-adjacent pair of vertices and a common neighbour.

Otherwise, the structure is as Claim 4(a) and (b) suggest. Find all $b \in B$ only adjacent to the multi-vertex part with one non-neighbour in that part in $O(n+m)$ time, by iterating over the adjacency lists of the vertices in B. Let this be the set of vertices B'. Then look for each vertex in B' whether all its neighbours in B have the same non-adjacency in the multi-vertex part in $O(n+m)$ time. If there is a vertex $b \in B'$ that meets this requirement, and G is (P_2+2P_1)-free, then this is a witness for diameter-3 shortest path: b is non-adjacent to one vertex $c \in C$ in a multi-vertex part of $G[C]$, which is the only part it is adjacent to, and $N(b) \cap N(c) = \emptyset$. So the distance from b to c is at least (and at most) 3. To verify, execute a BFS from any single one of these vertices. If a shortest path is found of length 4 or longer, then return that G is not (P_2+2P_1)-free. If a length-3 shortest path is found, then return it. Otherwise, there is no length-3 shortest path in G from B to C, if G is (P_2+2P_1)-free.

If in none of the above cases a confirmation for a shortest path of length 3 or 4 was found, and the graph is (P_2+2P_1)-free and not a clique, then the diameter of G must be 2. Return some non-adjacent pair with some common neighbour as a shortest path in $O(n+m)$ time. □

Note that any $3P_1$-free graph is (P_2+2P_1)-free.

Corollary 2. *Given a $3P_1$-free graph G, we can compute the diameter of G in $O(n+m)$ time.*

2.2 (P_4+P_1)-free graphs

We show that we can decide whether the diameter of a (P_4+P_1)-free graph is equal to $d_{\max}$ in $O(n+m)$ time. The proof will identify all possibilities of a diameter-4 path occurring in relation to a BFS from an arbitrary vertex. Luckily, most cases reduce to some other case in the proof, and algorithmically speaking, only a few cases require algorithmic computation.

Theorem 7 (♠). *Given a (P_4+P_1)-free graph G, we can decide whether the diameter of G is equal to $d_{\max}=4$ in $O(n+m)$ time.*

Proof (Sketch). Let $G=(V,E)$ be a (connected) (P_4+P_1)-free graph. Indeed, $d_{\max}=4$. We view the structure of G from a BFS from an arbitrary vertex u. Let $C = V \setminus N[u]$ and denote $B = N(u)$. If $C=\emptyset$ or $B=\emptyset$, then the diameter

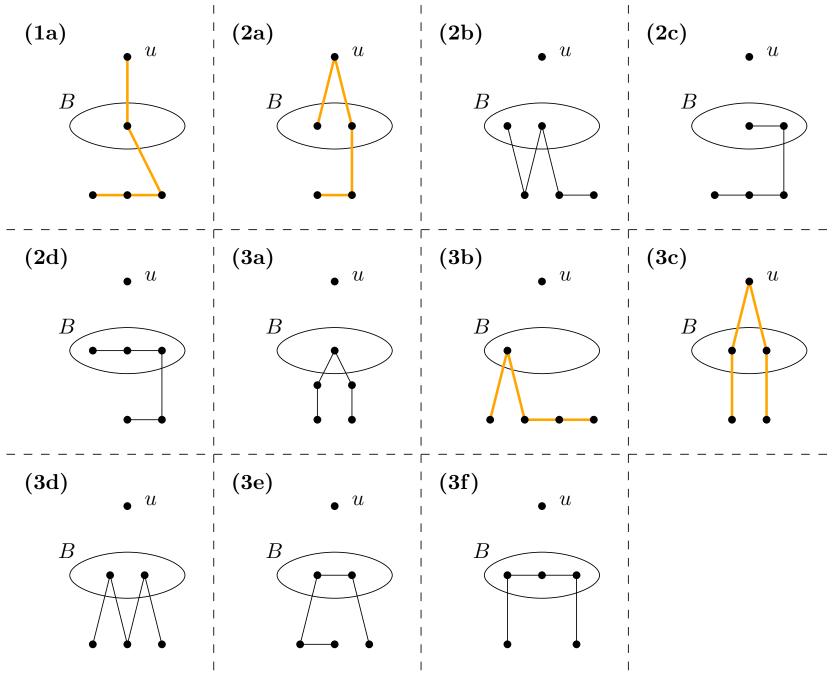

Fig. 2. An illustration of the types of a diameter-4 shortest path appearing in a $(P_4 + P_1)$-free graph, with respect to the vertex u and its neighbourhood $N(u) = B$. Only the highlighted types **(1a)**, **(2a)**, **(3b)**, **(3c)** require algorithmic computation.

of G is at most 2, so assume this is not the case. Note that both sets can be identified during the BFS from u with no overhead. Moreover, $G[C]$ is P_4-free. We use the convention that $b_i \in B$ and $c_i \in C$.

We first list all possibilities for a diameter-4 shortest path to exist with respect to its structure; see Figure 2 for an illustration and ♠ for the full case distinction. We will call each such possibility a 'type'. We show that we can decide whether the diameter of G is 4 in time $O(n + m)$ by resolving each type. Algorithmically speaking, only types **(1a)**, **(2a)**, **(3b)**, and **(3c)** will require computation to find diametral paths corresponding to them (highlighted in the figure). We will show that all other types are either covered by these computations, or are non-existent in G. As illustrations, we show **(1a)**, **(2a)**, and **(2b)** all with proof of correctness, and we give the algorithm for **(3c.1)**, a part of the proof for type **(3c)**; all remaining types and full proofs are discussed in ♠.

(1a) $\langle u, b, c_1, c_2, c_3 \rangle$ This type is identified by the initial BFS from u if and only if it occurs in G.

(2a) $\langle b_2, u, b_1, c_1, c_2 \rangle$ For this type, let us further partition C into sets C' and D, where D consists of the vertices with no neighbours in B (i.e. the vertices at distance 3 from u), and $C' = C \setminus D$. It must be that $c_2 \in D$ for a diametral path of type **(2a)** to exist, otherwise, $\langle c_2, b_i, u, b_2\rangle$ is a shorter path for some $b_i \in B$. So rename the vertex c_2 as $d = c_2 \in D$ and look for a path $\langle b_2, u, b_1, c_1, d\rangle$. Further partition C' into C_1 and C_2, where C_1 are the vertices with edges towards D and $C_2 = C' \setminus C_1$. This partitioning can be done during the BFS from u, or alternatively using another linear pass over all vertices and edges. Note that every vertex in D has at least one neighbour in C_1; otherwise, a diametral path of type **(1a)** exists in G, and we are done. Let us first describe the algorithmic steps necessary for this type.

(2a.algorithm) Find a vertex $d \in D$ with the smallest degree with respect to C_1, and execute a BFS from d. If a distance-4 vertex is found, return that the diameter of G is 4. If we do not find a vertex at distance 4, no distance-4 diametral path of type **(2a)** exists in G.

We next prove correctness of **(2a.algorithm)**. We prove correctness when $G[D]$ is not connected in **(2a.1)** and correctness when $G[D]$ is connected in **(2a.2)**. To do this, we analyse the structure of G under the assumption that a diametral path of type **(2a)** exists, to conclude the structure of G must then be 'simple' in some way, to the extent that the above algorithmic steps suffice.

(2a.1) Assume $G[D]$ is not connected. We first prove that every vertex in C_1 is complete to D. Assume for sake of contradiction that there is a $c \in C_1$ which is not complete to D. Let $d' \in D$ be a non-neighbour of c. Let $d \in D$ be some neighbour of c, which exists because $c \in C_1$. Now $\langle d, c, b, u\rangle$ is an induced P_4 for some $b \in B$ which exists by definition of C_1. But then d' must be a neighbour of d; otherwise, it would induce a $P_4 + P_1$. We see that every neighbour of c is adjacent to every non-neighbour of c in D. We get a contradiction with the assumption that $G[D]$ is not connected.

So, every vertex in C_1 must be complete to D. But then, from the viewpoint of shortest paths, for any $b \in B$, the distances from all $d \in D$ to b must be equal, as the shortest path to b must go through some vertex of C_1 in the first step, and C_1 is complete to D. Hence, if the BFS that **(2a.algorithm)** executes does not find a length-4 diametral path, no diametral path of type **(2a)** exists in G.

(2a.2) Assume $G[D]$ is connected. Then $G[C_1 \cup D]$ is a connected cograph, so it has diameter at most 2. Let $B_1 \subseteq B$ be the vertices of B with neighbours in C_1. Vertices in $B_2 = B \setminus B_1$ have no neighbours in C_1. Every vertex in B_1 has distance at most 3 to any $d \in D$, as the diameter of $G[C_1 \cup D]$ is at most 2. So, for a shortest path of type **(2a)** $\langle b_2, u, b_1, c_1, d\rangle$ we get $b_2 \in B_2$ and $b_1 \in B_1$.

Let us call a pair (d, b_2) with $d \in D$, $b_2 \in B_2$ 'good' if d has distance 4 to b_2, and 'bad' when d has distance at most 3 to b_2. Assuming a diametral path of form $\langle b_2, u, b_1, c_1, d\rangle$ exists in G, with $b_2 \in B_2$, $b_1 \in B_1$, $c_1 \in C_1$, $d \in D$, it is clear that (d, b_2) is good. Assume that we also have that (d', b_2) is bad for some $d \neq d' \in D$. Then d' has distance exactly 3 to b_2, as b_2 has no neighbour in C_1. Let c_1' be the neighbour of d' on some distance-3 path from d' to b_2. Then $c_1 \neq c_1'$; otherwise, d has distance 3 to b_2. Then the shortest path from d' to b_2 is of the form $\langle d', c_1', c_2', b_2\rangle$ with $c_2' \in C_2$ (case **(2a.2.1)**), or $\langle d', c_1', b_1', b_2\rangle$

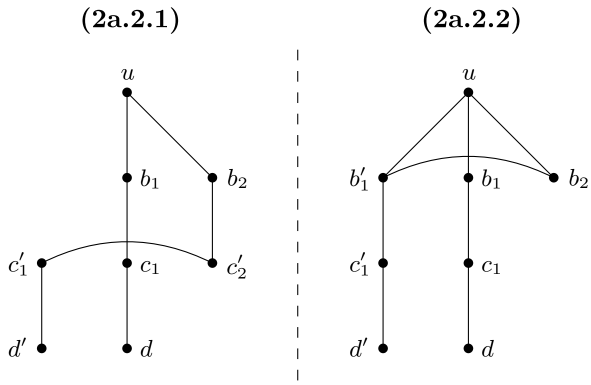

Fig. 3. Structure for type **(2a)** with respect to a $d \in D$ and a $d' \in D$ for which (d, b_2) is good and (d', b_2) is bad for $b_2 \in B_2$. The path from d' to b_2 goes through either some $c'_2 \in C_2$ (left, **(2a.2.1)**) or some $b'_1 \in B_1$ (right, **(2a.2.2)**).

with $b'_1 \in B_1$, $b'_1 \neq b_1$ (case **(2a.2.2)**). See Figure 3 for an illustration of both scenarios.

(2a.2.1) The shortest path from d' to b_2 is of the form $\langle d', c'_1, c'_2, b_2\rangle$ with $c'_2 \in C_2$. Note that $(c_1, u), (c'_1, u) \notin E$ by definition of C, and $(c_1, b_2), (c'_1, b_2) \notin E$ by definition of B_2, and $(b_1, b_2), (d, c'_1), (c_1, c'_2), (d, c'_2) \notin E$ as (d, b_2) is good. But then $(c'_1, c_1) \in E$ or $(c'_1, b_1) \in E$; otherwise, $\langle c_1, b_1, u, b_2\rangle + c'_1$ is an induced $P_4 + P_1$ in the graph.

- If $(c'_1, c_1) \in E$, then $\langle c'_2, c'_1, c_1, d\rangle + u$ is an induced $P_4 + P_1$ in the graph; u is non-adjacent to all of c'_2, c'_1, c_1, d by definition of C and D.
- If $(c'_1, b_1) \in E$ then $\langle b_2, u, b_1, c'_1\rangle + d$ is an induced $P_4 + P_1$ in the graph; $(c'_1, u) \notin E$ by definition of C, and (d, b_1), (d, b_2), $(d, u) \notin E$ by definition of D.

Hence, we get a contradiction, and this case cannot occur in G.

(2a.2.2) The shortest path from d' to b_2 is of the form $\langle d', c'_1, b'_1, b_2\rangle$ with $b'_1 \in B_1$, $b'_1 \neq b_1$. Note that $(c_1, u) \notin E$ by definition of C, $(d', b_1), (d', b_2), (d', u) \notin E$ by definition of D, and $(b_1, b_2), (c_1, b_2) \notin E$, as (d, b_2) is good. But then $(d', c_1) \in E$: otherwise, $\langle c_1, b_1, u, b_2\rangle + d'$ is an induced $P_4 + P_1$ in the graph.

Say that d has some other neighbour $c''_1 \in C_1$, so $(c''_1, b''_1) \in E$ for some $b''_1 \in B_1$ (possibly $b''_1 = b_1$), but $(b''_1, b_2) \notin E$ because (d, b_2) is good. Then, c''_1 can fulfil the role of c_1 in the above analysis, so it must be that $(d', c''_1) \in E$. From this analysis we can conclude that, for any $b_2 \in B_2$, if $d \in D$ is such that (d, b_2) is good, and $d' \in D$ is such that (d', b_2) is bad, then it must be that $N(d) \cap C_1 \subset N(d') \cap C_1$. Hence, if some $d \in D$ has minimum degree to C, it may be good.

We now have the tool to prove correctness of **(2a.algorithm)** for this case. If the BFS from the picked $d \in D$ finds a vertex at distance 4, then clearly a length-4 diametral path exists in G. The only risk is that we conclude there is no diametral path corresponding to this case even though it does exist. To this end, let $d \in D$ be the vertex picked by the algorithm, and assume (d, b) is bad for all $b \in B_2$. Assume to the contrary that there exist $d' \in D$, $b_2 \in B_2$ which are at distance 4 in a path of type **(2a)**. Then (d', b_2) is good. So, by the analysis above, it must be that $N(d') \cap C_1 \subset N(d) \cap C_1$, which contradicts the assumption that d was picked to have the minimal size neighbourhood with respect to C_1. So all pairs (d, b_2) with $d \in D$, $b_2 \in B_2$ must be bad, and we are correct to conclude that a diametral path of type **(2a)** does not exist.

(2b) $\langle b_1, c_1, b_2, c_2, c_3 \rangle$ As b_1 and b_2 are both adjacent to u, if type **(2b)** exists in G, $\langle b_1, u, b_2, c_2, c_3 \rangle$ is also a shortest path of distance 4 from b_1 to c_3, which is a distance-4 path of type **(2a)**. The algorithm for type **(2a)** finds a diametral path of length 4 or concludes that no diametral path of type **(2a)** exists in G, which also rules out that a diametral path of type **(2b)** exists in G.

(3c) $\langle c_1, b_1, u, b_2, c_2 \rangle$ We only treat one subcase: **(3c.1)** Every $b \in B$ is complete or anti-complete to every component of $G[C]$. It can be checked in $O(n + m)$ time whether we are in this case. We give the algorithm to solve this case, and the proof of correctness is deferred (♠).

(3c.1.algorithm) For every component of $G[C]$, delete all-but-one vertex in $O(n+m)$ time, each component is a twin class with respect to B. Let the resulting set of vertices be C'. Identify twins within B with respect to their neighbourhood in C', which can be done in linear time by Corollary 1. We identify every 'class' of vertices with the same neighbourhood in C' with single vertices. Let B' be the set of vertices corresponding to classes. The neighbourhoods of the vertices in B' are the union of all neighbourhoods of vertices in the class, which can be computed in $O(n+m)$ time total (♠). Call the resulting graph $G' = (V', E')$ with $B', C' \subseteq V'$. In G', check that $G'[B']$ consists of two non-empty disjoint cliques B_1, B_2, where B_1 and B_2 are anti-complete, with possibly a third clique X complete to $B' \setminus X$. This check takes $O(n+m)$ time by inspecting the neighbourhoods of all vertices in B'. If $G'[B']$ does not have this structure, then return that there is no diametral path of type **(3c)**. If it does, then return that a diametral path of type **(3c)** exists if there are vertices $c_1, c_2 \in C'$ with $N(c_1) = B_1$ and $N(c_2) = B_2$. This can be checked in linear time by inspecting the edge lists of vertices $c \in C'$.

The correctness of **(3c.1.algorithm)** follows from structural analysis on the graph G'. However, G' must be treated with care; it is not necessarily a (P_4+P_1)-free graph. However, the fact that G is (P_4+P_1)-free leads to a strong structural property: for every non-edge $(b'_1, b'_2) \notin E'$ in $G'[B']$, it holds that b'_1, b'_2 together dominate C'. Such a non-edge must exist if a path of type **(3c)** exists in G.

The remaining cases and correctness are treated in the full version (♠). □

3 Conclusion and Discussion

We analysed the complexity of computing the diameter of H-free graphs. For several H-free graph classes $\mathcal{G}$ where H is a linear forest, we found linear-time algorithms for solving $d_{\max}(\mathcal{G})$-Diameter. We conjecture that this generalizes to a broader set of H-free graphs. In particular, this conjecture emphasizes that H should include some set of isolated vertices, as it seems that in our algorithms, the presence of a P_1 in H helps out in structural analysis.

Conjecture 1. Let $\mathcal{G}$ be the class of $rP_t + sP_1$-free graphs, where $r, s, t \geq 1$. Then $d_{\max}(\mathcal{G})$-Diameter can be solved in $O(n+m)$ time.

Note that all our results support Conjecture 1. A linear-time algorithm for the only open case with $d_{\max} = 4$, which is for $H = P_3 + P_2$, would suggest Conjecture 1 is not the full truth, if it holds true at all. In the full discussion and conclusion (♠), we give two more conjectures in an attempt to reveal the underlying patterns in our results.

The main open problem posed by our work is whether our conjectures hold true. The smallest open cases for solving $d_{\max}(\mathcal{G})$-Diameter are algorithms for the classes of H-free graphs with $H = 5P_1$ ($d_{\max} = 7$), $H = 2P_2 + 2P_1$ ($d_{\max} = 7$), $H = P_3 + P_2$ ($d_{\max} = 4$), $H = P_4 + 2P_1$ ($d_{\max} = 6$), or $H = P_5 + P_1$ ($d_{\max} = 5$), and a hardness result for the class of P_6-free graphs. The specific case of $H = P_4 + 2P_1$ is interesting because it is the only graph H for which we have no result.

Settling the following open question would form a complete dichotomy for $(2P_2 + P_1)$-free graphs:

Given a $(2P_2 + P_1)$-free graph G, can we decide in $O(n+m)$ time whether the diameter of G is equal to 4?

We make progress for Diameter computation on H-free graphs, but do not settle its complexity completely, so we ask:

When $H = 4P_1$, $H = P_2 + 3P_1$, $H = P_3 + 2P_1$, $H = P_4 + 2P_1$, or $H = P_4 + P_1$, can we solve the Diameter *problem on connected H-free graphs in $O(n+m)$ time?*

Acknowledgments. We thank the reviewers for their helpful comments and suggestions.

References

1. Abboud, A., Grandoni, F., Williams, V.V.: Subcubic equivalences between graph centrality problems, APSP, and diameter. ACM Trans. Algorithms **19**(1), 3:1–3:30 (2023)
2. Abboud, A., Mozes, S., Weimann, O.: What else can Voronoi diagrams do for diameter in planar graphs? In: Proc. ESA 2023. LIPIcs, vol. 274, pp. 4:1–4:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2023)
3. Abboud, A., Williams, R.R., Yu, H.: More applications of the polynomial method to algorithm design. In: Proc. SODA 2015. pp. 218–230. SIAM (2015)
4. Abboud, A., Williams, V.V., Wang, J.R.: Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In: Proc. SODA 2016. pp. 377–391. SIAM (2016)

5. Backurs, A., Roditty, L., Segal, G., Williams, V.V., Wein, N.: Towards tight approximation bounds for graph diameter and eccentricities. In: Proc. STOC 2018. pp. 267–280. ACM (2018)
6. Brandstädt, A., Chepoi, V., Dragan, F.F.: The algorithmic use of hypertree structure and maximum neighbourhood orderings. Discret. Appl. Math. **82**(1-3), 43–77 (1998)
7. Bringmann, K., Husfeldt, T., Magnusson, M.: Multivariate analysis of orthogonal range searching and graph distances. Algorithmica **82**(8), 2292–2315 (2020)
8. Cabello, S.: Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. ACM Trans. Algorithms **15**(2), 21:1–21:38 (2019)
9. Chan, T.M., Williams, R.R.: Deterministic APSP, Orthogonal Vectors, and more: Quickly derandomizing Razborov-Smolensky. ACM Trans. Algorithms **17**(1), 2:1–2:14 (2021)
10. Chechik, S., Larkin, D.H., Roditty, L., Schoenebeck, G., Tarjan, R.E., Williams, V.V.: Better approximation algorithms for the graph diameter. In: Proc. SODA 2014. pp. 1041–1052. SIAM (2014)
11. Chepoi, V., Dragan, F.F., Estellon, B., Habib, M., Vaxès, Y.: Notes on diameters, centers, and approximating trees of delta-hyperbolic geodesic spaces and graphs. Electron. Notes Discret. Math. **31**, 231–234 (2008)
12. Chepoi, V., Dragan, F.F., Habib, M., Vaxès, Y., Alrasheed, H.: Fast approximation of eccentricities and distances in hyperbolic graphs. JGAA **23**(2), 393–433 (2019)
13. Corneil, D.G., Dragan, F.F., Habib, M., Paul, C.: Diameter determination on restricted graph families. Discret. Appl. Math. **113**(2-3), 143–166 (2001)
14. Corneil, D.G., Dragan, F.F., Köhler, E.: On the power of BFS to determine a graph's diameter. Networks **42**(4), 209–222 (2003)
15. Coudert, D., Ducoffe, G., Popa, A.: Fully polynomial FPT algorithms for some classes of bounded clique-width graphs. ACM Trans. Algorithms **15**(3), 33:1–33:57 (2019)
16. Cygan, M., Gabow, H.N., Sankowski, P.: Algorithmic applications of baur-strassen's theorem: Shortest cycles, diameter, and matchings. J. ACM **62**(4), 28:1–28:30 (2015)
17. Dragan, F.F.: Centers of graphs and the helly property. Ph. D. Thesis, Moldova State University (1989)
18. Dragan, F.F.: HT-graphs: centers, connected r-domination and steiner trees. Comput. Sci. J. Moldova **1**(2), 64–83 (1993)
19. Dragan, F.F.: Dominating cliques in distance-hereditary graphs. In: Proc. SWAT 1994. LNCS, vol. 824, pp. 370–381. Springer (1994)
20. Dragan, F.F., Ducoffe, G.: α_i-metric graphs: Radius, diameter and all eccentricities. In: Proc. WG 2023. LNCS, vol. 14093, pp. 276–290. Springer (2023)
21. Dragan, F.F., Ducoffe, G., Guarnera, H.M.: Fast deterministic algorithms for computing all eccentricities in (hyperbolic) Helly graphs. In: Proc. WADS 2021. LNCS, vol. 12808, pp. 300–314. Springer (2021)
22. Dragan, F.F., Guarnera, H.M.: Eccentricity function in distance-hereditary graphs. Theor. Comput. Sci. **833**, 26–40 (2020)
23. Dragan, F.F., Habib, M., Viennot, L.: Revisiting radius, diameter, and all eccentricity computation in graphs through certificates. CoRR **abs/1803.04660** (2018)
24. Dragan, F.F., Nicolai, F.: LexBFS-orderings of distance-hereditary graphs with application to the diametral pair problem. Discret. Appl. Math. **98**(3), 191–207 (2000)
25. Dragan, F.F., Nicolai, F., Brandstädt, A.: LexBFS-orderings and powers of graphs. In: Proc. WG 1996. pp. 166–180. Springer (1997)

26. Duan, R., Wu, H., Zhou, R.: Faster matrix multiplication via asymmetric hashing. In: Proc. FOCS 2023. pp. 2129–2138. IEEE (2023)
27. Ducoffe, G.: Beyond Helly graphs: The diameter problem on absolute retracts. In: Proc. WG 2021. LNCS, vol. 12911, pp. 321–335. Springer (2021)
28. Ducoffe, G.: The diameter of AT-free graphs. J. Graph Theory **99**(4), 594–614 (2022)
29. Ducoffe, G.: Optimal centrality computations within bounded clique-width graphs. Algorithmica **84**(11), 3192–3222 (2022)
30. Ducoffe, G.: Distance problems within Helly graphs and k-Helly graphs. Theor. Comput. Sci. **946**, 113690 (2023)
31. Ducoffe, G., Dragan, F.F.: A story of diameter, radius, and (almost) Helly property. Networks **77**(3), 435–453 (2021)
32. Ducoffe, G., Habib, M., Viennot, L.: Diameter computation on H-minor free graphs and graphs of bounded (distance) VC-dimension. In: Proc. SODA 2020. pp. 1905–1922. SIAM (2020)
33. Duraj, L., Konieczny, F., Potepa, K.: Better diameter algorithms for bounded VC-dimension graphs and geometric intersection graphs. CoRR **abs/2307.08162** (2023)
34. Evald, J., Dahlgaard, S.: Tight hardness results for distance and centrality problems in constant degree graphs. CoRR **abs/1609.08403** (2016)
35. Farley, A.M., Proskurowski, A.: Computation of the center and diameter of outerplanar graphs. Discret. Appl. Math. **2**(3), 185–191 (1980)
36. Gawrychowski, P., Kaplan, H., Mozes, S., Sharir, M., Weimann, O.: Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{o}(n^{5/3})$ time. SIAM J. Comput. **50**(2), 509–554 (2021)
37. Impagliazzo, R., Paturi, R.: On the complexity of k-SAT. J. Comput. Syst. Sci. **62**(2), 367–375 (2001)
38. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. Syst. Sci. **63**(4), 512–530 (2001)
39. Johnson, M., Martin, B., Oostveen, J.J., Pandey, S., Paulusma, D., Smith, S., van Leeuwen, E.J.: Complexity framework for forbidden subgraphs. CoRR **abs/2211.12887** (2022)
40. Olariu, S.: A simple linear-time algorithm for computing the center of an interval graph. Int. J. Comput. Math. **34**(3-4), 121–128 (1990)
41. Oostveen, J.J., Paulusma, D., van Leeuwen, E.J.: The complexity of diameter on H-free graphs. CoRR **abs/2402.16678** (2024)
42. Roditty, L., Williams, V.V.: Fast approximation algorithms for the diameter and radius of sparse graphs. In: Proc. STOC 2013. pp. 515–524. ACM (2013)
43. Shoshan, A., Zwick, U.: All pairs shortest paths in undirected graphs with integer weights. In: Proc. FOCS 1999. pp. 605–615. IEEE (1999)
44. Weimann, O., Yuster, R.: Approximating the diameter of planar graphs in near linear time. ACM Trans. Algorithms **12**(1), 12:1–12:13 (2016)
45. Zwick, U.: All pairs shortest paths using bridging sets and rectangular matrix multiplication. J. ACM **49**(3), 289–317 (2002)