Computing Giant Graph Diameters

Peter Damaschke

Department of Computer Science and Engineering
Chalmers University, 41296 Göteborg, Sweden
ptr@chalmers.se

Abstract. This paper is devoted to the fast and exact diameter computation in graphs with n vertices and m edges, if the diameter is a large fraction of n. We give an optimal $O(m+n)$ time algorithm for diameters above $n/2$. The problem changes its structure at diameter value $n/2$, as large cycles may be present. We propose a randomized $O(m + n \log n)$ time algorithm for diameters above $(1/3 + \epsilon)n$ for constant $\epsilon > 0$.

1 Introduction

Computing distances and shortest paths is one of the fundamental graph problems. The diameter of an undirected graph is the maximum distance between any two vertices. In a graph with n vertices and m edges of unit length, all distances from a single vertex (single-source shortest paths, SSSP) can be computed by breadth-first-search (BFS) in $O(n+m)$ time, and all pairwise distances (all-pairs shortest paths, APSP) can therefore be obtained in $O(nm)$ time by solving n times SSSP. Trivially, this also yields the diameter, but it was a longstanding open problem whether the diameter can be computed significantly faster than via APSP, see [1, 14] for results. Many results are also known for diameter computation in special graph classes [5, 6, 8, 13] and fast approximation of the diameter [2–4, 11, 13]. This bibliography is certainly far from being complete. Other related lines of research that we cannot survey here include faster APSP computation in special graph classes, and experimental studies of diameter computations in real-world graphs.

Instead of the graph structure one may also restrict the range of diameters. As discussed in [5], the problem of distinguishing between graphs of diameter 2 and 3, already for the special class of split graphs, is as hard as the disjoint sets problem (deciding whether a given set family contains two disjoint sets) and is therefore unlikely to have a subquadratic algorithm. In the present paper we look at the other end: graphs with "giant" diameters close to the number n of vertices. (The word is borrowed from the giant components of random graphs.) Whereas most real-world networks have small diameters, chain-like structures may appear as well in various contexts (chain molecules, connections between two fixed sites in a network, etc.).

Contributions. First we give an $O(m + n)$ time algorithm for diameters above $n/2$. One can think of different approaches, e.g., similar to diameter computation in trees. Our approach is based on separators (articulation points in this case), and removal of irrelevant subgraphs. Moreover, it is not necessary to know in advance that the given graph has a large diameter. Admittedly we use quite a number of lemmas to prepare this result, but we want to point out all single steps, in the hope that future research can generalize them to larger separators and smaller diameters. We also show that the $O(n+m)$ time bound cannot be improved (say, to $O(n)$ time) under plausible assumptions on the graph representation. While our solution for diameters larger than $n/2$ works with articulation points, we observe some "phase transition" just below $n/2$: A graph with such a diameter may have a giant geodesic cycle, hence qualitatively different methods are needed to "choose the correct half cycle" that yields the diameter. For this purpose we define an auxiliary problem that might be of independent interest. Its solution is applied in a randomized $O(m+n\log n)$ time algorithm for diameters above $n/3$. Note that this bound is linear in the graph size if $m > n\log n$.

2 Preliminaries

Our graphs $G=(V,E)$ are undirected, unweighted, and connected, and have n vertices and m edges. A path joining two vertices u and v is denoted $u-v$, if the inner vertices are clear from context or irrelevant. The distance $d_G(u,v)$, or simply $d(u,v)$, is the length, i.e., number of edges, of a shortest $u-v$ path. A shortcut to a subgraph H of G is a path in G that connects two vertices u and v from H, but is shorter than $d_H(u,v)$. We call H a geodesic subgraph if H has no shortcuts in G. In particular, a geodesic path P is a shortest path connecting its end vertices. The diameter of G is $diam(G) := \max\{d(u,v)|\, u,v \in V\}$. Hence a longest geodesic path in G is a path of length $diam(G)$. We use the abbreviation $\delta := diam(G)/n$. Note that a cycle C is geodesic if, for any two vertices $u,v \in C$, their smaller distance (of at most $\frac{1}{2}|C|$) on C equals $d(u,v)$.

With respect to a root vertex r we refer to the sets $N_i(r) := \{v|\, d(u,v)=i\}$ as layers, and the depth of G is defined by $\max\{d(r,v)|\, v \in V\}$. Depth and layers can be computed using breadth-first-search (BFS).

To avoid heavy notation and technicalities we may neglect additive constants in arithmetic expressions, as well as rounding of fractional numbers to integers, as long as this does not affect asymptotic statements for large graphs.

For $U \subset V$ we denote by $G-U$ the graph that remains when the vertices of U and all incident edges are removed from G. If $U=\{u\}$, we write $G-u$ for $G-U$. A separator is a vertex set $S \subset V$ such that $G-S$ is disconnected. An articulation point is just a separator of size 1, that is, a vertex u such that $G-u$ is disconnected. A block is a biconnected graph, that is, a graph without articulation points. The block-cut tree of G has a node for every articulation point, and a node for every block (biconnected component) without the articulation points therein. The block-cut tree has edges between adjacent articulation points, and between those articulation points and blocks where G has edges.

A hair in a graph is a path H such that one end vertex of H has degree 1, all inner vertices have degree 2, and the other end vertex has degree larger than

2. We can think of a hair as a simple path that is dangling at the rest of the graph. In particular, a hair is a geodesic path.

We say that a vertex v is between vertices u and w, in symbols $B(u, v, w)$, if the triangle inequality degenerates to the equation $d(u, w) = d(u, v) + d(v, w)$.

We tacitly use some elementary properties listed here: Any subpath of a geodesic path is geodesic. If we replace any subpath of a geodesic path with another geodesic (sub)path between the same two vertices, then the entire path remains geodesic. Any three vertices u, v, w that appear in this order on a geodesic path satisfy $B(u, v, w)$. Conversely, if $B(u, v, w)$ holds true, then any concatenation of two geodesic $u - v$ and $v - w$ paths is a geodesic $u - w$ path.

3 Diameters Larger than Half the Size

First we study the largest diameters, more precisely, the case $\delta > 1/2$. We show that this case can be solved in linear time. Our approach works with articulation points, and (in Lemma 3) pruning of irrelevant vertices. Lemma 2 below also holds for general graphs.

Lemma 1. *Suppose that $\delta = 1/2 + h$, and let P be any longest geodesic path. Then there exists a vertex $u \in P$ which is an articulation point of G and divides P in two subpaths of length at least hn each.*

Proof. Let v be an end vertex of P. Clearly, the number of vertices not in P is $(1/2 - h)n$. Hence at least $2hn$ of the layers $N_i(v)$ contain a single vertex. Since edges cannot skip layers, every such single vertex u (except for $i = 0$ and possibly the last layer) is an articulation point of G and an inner vertex of P.

Specifically, consider an articulation point u that belongs to P and is as close as possible to the center of P. In the worst case, only $2hn$ articulation points are on P, and they form two subpaths of equal lengths at the ends of P. Still, an innermost articulation point u divides P in two paths the shorter of which has length at least hn. □

Lemma 2. *Consider an articulation point u of G, a connected component C of $G - u$, and a longest geodesic path P in G. Define $C_u := C \cup \{u\}$. Then one of these three cases applies: (a) P does not intersect C. (b) P intersects both C and $G - C_u$. (c) P is entirely in C_u.*
In case (b), the subpath P_u of P in C_u is a geodesic path connecting u with some vertex of C at maximum distance from u. Moreover, any such geodesic path in C_u may replace P_u in P, and the resulting path is again a longest geodesic path in G. Case (c) can be true only if C_u has at least δn vertices.

Proof. The case distinction is evident, as well as the assertion about case (c). The assertion about case (b) follows from two facts: P has maximum length, and no edges join any vertices of C and $G - C_u$. Hence the new subpath cannot lead to shortcuts to vertices outside C. □

Lemma 3. *Consider an articulation point u of G, a connected component C of $G - u$, and a longest geodesic path P in G. Suppose that P is not entirely in C (for instance, because C has fewer than δn vertices). Then it is safe to keep only one geodesic path from u to a farthest vertex v (with maximum $d(u, v)$) in C and remove all other vertices of C. That is, this removal retains some longest geodesic path in G.*

Proof. By assumption, case (c) of Lemma 2 does not apply to C. If case (a) applies, then the assertion is vacuously true. If case (b) applies, then the assertion follows from the property mentioned in Lemma 2: Since any geodesic path from u to a farthest vertex v can be used, we need to keep only one. □

As a consequence of the previous lemmas we can already settle one case:

Lemma 4. *Suppose that $\delta > 1/2$. Let u be an articulation point of G such that every connected component of $G - u$ has fewer than $n/2$ vertices. Then G has a longest geodesic path P composed of two subpaths that connect u with the farthest vertices in two distinct connected components of $G - u$ with the two largest depths. (Here, depth is understood with respect to the root u, and ties are broken arbitrarily if some depths are equal.)*

Proof. P has the claimed shape due to Lemma 3. Since P has the maximum length among all geodesic paths, the two connected components that intersect P must also have the largest depths. □

The next lemma addresses some routine preprocessing.

Lemma 5. *Given a graph G, we can determine, in $O(n+m)$ time, the set A of all articulation points u, the block-cut tree of G, and the vertex numbers of all connected components of all graphs $G - u$ ($u \in A$).*

Proof. In $O(n + m)$ time one can find all articulation points of G [9, 12], and furthermore construct the block-cut tree T straightforwardly. We declare an arbitrary node of T the root and compute, by bottom-up summation in the rooted tree T, the number of vertices (that is, original vertices of G) below every edge of T. From these numbers we get the vertex numbers of all connected components of $G - u$, for all articulation points u, in $O(n + m)$ time in total: In particular, note that one edge from any articulation point u except the root goes upwards in the rooted tree, and the size of the corresponding component is $n - 1$ minus the sum of sizes of all other connected components of $G - u$ being below u in the rooted tree. □

Now we can either reduce an instance of our problem in linear time to an equivalent instance with a simple structure, or solve the problem.

Lemma 6. *In a graph G with $\delta > 1/2$ we can, in $O(n + m)$ time, either compute a longest geodesic path of G, or extract an induced subgraph of G that still contains a longest geodesic path of G and consists of only one block with hairs.*

Proof. We do computations as in Lemma 5. If, for an articulation point u, every connected component of $G-u$ has fewer than $n/2$ vertices, then we find a longest geodesic path by Lemma 4 in $O(n+m)$ time, by using BFS with root u.

The other case is that, for every articulation point u, one connected component of $G-u$ has at least $n/2$ vertices. Assume that the block-cut tree T has two or more blocks. Then there exists an articulation point u on the path of T between any two blocks. But now Lemma 3 applies to the connected components of $G-u$ except the largest one. Thus we can replace them all with one longest geodesic path from u into these components, ending now in a new leaf of T. In particular, we get rid of at least one block.

We repeat this procedure until only one block with hairs remains. The depths and hairs are computed by BFS, where we can append any previously computed hair as a whole, if BFS reaches its (non-leaf) start vertex. Thus all changes affect pairwise disjoint parts of T, thus the process costs $O(n+m)$ time in total. □

In order to compute a longest geodesic path in arbitrary graphs with $\delta > 1/2$ it remains to treat the graphs as produced in Lemma 6, consisting of one block with hairs. Note that still $\delta > 1/2$, since the number of vertices has not increased. Now we also use the quantitative part of Lemma 1.

Lemma 7. *In a graph G with $\delta > 1/2$ consisting of one block with hairs, some longest geodesic path begins at one of the two longest hairs (where ties are broken arbitrarily if some hair lengths are equal).*

Proof. Lemma 1 implies for this special type of graph that any longest geodesic path P must begin with a hair of length at least hn. We define factors h_i such that $h_1 n \geq h_2 n \geq \ldots$ are the hair lengths in descending order, and we let $H_1, H_2 \ldots$ denote the hairs in this order (not including their last articulation points that belong to the block).

If P does not begin with H_1, then P is a longest geodesic path in the graph $G_1 := G - H_1$, thus in a graph with $n_1 := (1-h_1)n$ vertices and with diameter $(\frac{1}{2}+h)n = (\frac{1}{2}-\frac{1}{2}h_1+\frac{1}{2}h_1+h)n = \frac{1}{2}n_1 + (\frac{1}{2}h_1+h)n$. Since Lemma 1 also holds for G_1, we conclude that P must begin with a hair of length at least $(\frac{1}{2}h_1+h)n$, thus $\frac{1}{2}h_1 + h \leq h_2$. If P does not begin with H_2 either, then P is a longest geodesic path in $G_2 := G_1 - H_2$, thus in a graph with $n_2 := (1-h_1-h_2)n$ vertices and, by a similar calculation, with diameter $\frac{1}{2}n_2 + (\frac{1}{2}(h_1+h_2)+h)n$. The same reasoning as above implies $\frac{1}{2}(h_1+h_2) + h \leq h_3$. This contradicts $h_1 \geq h_2 \geq h_3$. Thus, P must begin with H_1 or H_2. □

This yields the final result of the section.

Theorem 1. *In a graph G with $\delta > 1/2$ we can find some longest geodesic path, and thus compute diam(G), in $O(n+m)$ time.*

Proof. We run the procedure from Lemma 6. If it yields a subgraph of the special form mentioned there, we start BFS from the two longest hairs and output the longest of the two geodesic paths, which is correct by Lemma 7. □

Corollary 1. *In a graph G we can decide whether $\delta > 1/2$, and in that case we can find some longest geodesic path, and thus compute $diam(G)$, altogether in $O(n + m)$ time.*

Proof. First we run an algorithm as in Theorem 1. (We remark that the following reasoning does not depend on the particular algorithm.) If it does not output a result, then $\delta \leq 1/2$. Otherwise, we test in $O(n + m)$ time whether the output path actually has a length above $n/2$ and is a geodesic path. This can be done by BFS from one end vertex, since BFS yields the distances from the root vertex. If the output passes the test, then $\delta > 1/2$. Conversely, if $\delta > 1/2$, then the test confirms it. □

4 Optimality of Linear Time (in the Number of Edges)

Graphs with $\delta > 1/2$ can still have a quadratic number $m = O(n^2)$ of edges. For instance, consider a path of length δn with a clique of $(1 - \delta)n$ vertices attached somewhere. One may suspect that we need not read all edges in dense subgraphs in order to compute $diam(G)$, since most of them cannot belong to a longest geodesic path. Therefore it is not obvious whether the time $O(n + m)$ is optimal. Perhaps one could solve the problem in $O(n)$ time? However, we will argue that $O(n + m)$ time is actually needed in the worst case, even for a good approximation, provided that graphs are given by adjacency lists where the vertices appear in no particular order. The idea is that $diam(G)$ can depend on the presence of single edges creating shortcuts, but they are hard to find between dense subgraphs. The crucial subproblem in pure form looks as follows.

Crossing Edge: Given is a graph on a vertex set $X \cup Y$, where $X \cap Y = \emptyset$. The graph is given by adjacency lists, where the vertices appear in no particular order, and the partitioning into X and Y is known. Find some edge xy with $x \in X$, $y \in Y$, or report that no such edge exist.

Note that the following lemma hinges on the cardinalities. It would not hold if, for instance, $|X| = k$ and $|Y| = 1$.

Lemma 8. *Any algorithm that solves* Crossing Edge *with* $|X| = |Y| = k$ *needs* $\Omega(k^2)$ *time in the worst case.*

Proof. We can think of any algorithm as a player that can only look up entries in the adjacency lists, whereas an adversary provides all information. This translates the problem into a game with the following rules. In each step, the player may choose an arbitrary vertex u, and the adversary returns one vertex v adjacent to u (meaning that the player reads v in u's adjacency list).

As we are proving a lower bound, we can give the player extra information: The adversary tells in advance that either none or two edges exist between X and Y, and all other edges are inside X or Y. The player also gets to know the degrees of all vertices, that is, the lengths of all adjacency lists. Now the player can examine the adjacency lists, thus learn the edges. After each step of the game, the adversary is even more helpful and removes not only v from u's adjacency list, but also u from v's adjacency list. Only the undetected edges are kept, and the degrees of u and v are reduced by 1.

It remains to specify an adversary strategy. Remember that the player's instantaneous knowledge is the degrees of all vertices of X and Y, respectively. We call a degree sequence (multiset of degrees) valid, if there exists a graph with that degree sequence. The adversary does not reveal the graph, but only valid degree sequences in both X and Y. Initially let all degrees be $k-1$, thus we have roughly k^2 edges, and the degree sequences are valid, as both subgraphs can be cliques. As long as there remains at least one edge in both X and Y, the player cannot distinguish whether these edges in X and Y exist, or instead two edges between X and Y joining the same four vertices. Whenever the player has chosen a vertex u, the adversary takes a vertex v from the same set ($v \in X$ if $u \in X$, and $v \in Y$ if $u \in Y$) such that the resulting degree sequence after subtracting 1 remains valid. Such a vertex v does always exist: Since the current degree sequence is valid, there exists a graph realizing it, and in such a graph there exists an edge uv that can be removed.

This shows that the player must empty one of X and Y, and therefore see $\Omega(k^2)$ edges, in order to decide whether some edges join X and Y. ◻

Proposition 1. Any algorithm that approximates the diameter of graphs with any fixed $\delta > 1/2$ within a factor better than 2 needs $\Omega(n+m)$ time in the worst case.

Proof. We construct a special graph G: We take a simple path P of length δn and attach two subgraphs with vertex sets X and Y at the ends of P, $|X| = |Y| = k := \frac{1}{2}(1-\delta)n$. They are chosen as in Lemma 8; in particular, we have $m = \delta n + \Theta(k^2) = \Theta(n^2)$ edges. If X and Y are connected directly by some edge, then $diam(G) = \frac{1}{2}\delta n$ rather than $diam(G) = \delta n$ "as expected". By Lemma 8, a shortcut between X and Y cannot be recognized or excluded without reading $\Omega(k^2) = \Omega(n+m)$ edges, as this problem is an instance of Crossing Edge. ◻

5 An Auxiliary Problem: Largest Mixed Sum

For $\delta \leq 1/2$, diameter computation cannot be based on articulation points any more, for the trivial reason that there exist graphs with diameter about $n/2$ but without any articulation points, such as the chordless cycle. We argue that $\delta = 1/2$ is a barrier in the sense that already for δ slightly below $1/2$, due to the possibility of long geodesic cycles and the lack of articulation points, it is inevitable for diameter calculation to solve a specific new subproblem.

To introduce and motivate this problem, consider the following special case of graphs. Let $H = (V, E)$ and $H' = (V', E')$ be two vertex-disjoint graphs with distinguished vertices $u, v \in V$ and $u', v' \in V'$. We connect u and u' by a path of some length ℓ larger than the diameters of H and H'. Similarly we connect v and v' by another path of length ℓ, being vertex-disjoint to the first path. The graph G constructed in this way is, roughly speaking, a geodesic cycle with two subgraphs H and H' attached at diametral positions. For any two vertices $w \in V$ and $w' \in V'$ we have $d(w,w') = \ell + \min\{d(w,u)+d(w',u'), d(w,v)+d(w'v')\}$, since one of the paths $u-u'$ or $v-v'$ must be chosen. (Distances are meant with respect to G.) Define $s := d(u,v)$ and $s' := d(u',v')$. Note that G has a longest geodesic cycle (in general not uniquely determined) of length $2\ell + s + s'$. Any geodesic path that starts outside $V \cup V'$ is a subpath of some longest geodesic cycle and has therefore a length at most $\ell + \frac{1}{2}(s+s')$. Some geodesic path connecting H and H' can be longer, since a distance $d(w,w')$, as above, can be as large as $\ell + \frac{1}{2}(d(w,u)+d(w',u')+d(w,v)+d(w'v')) \geq \ell + \frac{1}{2}(s+s')$. (The two terms under "min" might be equal, and the triangle inequality holds.) Then we must find the maximum $d(w,w')$ to get the correct diameter. By abstracting from the graph problem and using the symbols

$$x := d(w,u),\ y := d(w,v),\ y' := d(w',u'),\ x' := d(w',v'),$$

we arrive at the following problem statement.

Largest Mixed Sum: We are given h pairs of numbers (x_i, y_i) and h' pairs of numbers (x'_j, y'_j), find two indices i and j so as to maximize $\min\{x_i + y'_j, y_i + x'_j\}$. We refer to the given pairs as h red and h' blue pairs, and we refer to the given numbers as coordinates. We can assume $h' \leq h$.

Observe that these values x, y and x', y' for all vertices w and w', respectively, can together be computed by four runs of BFS, in linear time in the number of edges of H and H'. From any identical pairs we keep only one copy. We say that a pair of numbers (a,b) is dominated by a pair (c,d) if $a \leq c$ and $b \leq d$. Within a given set of pairs, we call a pair non-dominated if that pair is not dominated by other pairs in the set.

Proposition 2. Largest Mixed Sum is solvable in $O(h \log h)$ time.

Proof. The subset of the non-dominated pairs in a set of h pairs, sorted by strictly ascending first coordinates (and thus by strictly descending second coordinates) can be computed in $O(h \log h)$ time: Sort the pairs by their first coordinates, scan this sequence, and maintain the sorted sequence of pairs being non-dominated so far. Since the second coordinates are decreasing there, for every new pair (a,b) we only have to find the correct place of b in the sequence by binary search, and then delete the current end of the sequence containing those pairs with second coordinates smaller than b.

An optimal solution to Largest Mixed Sum can always be formed by a red pair and a blue pair which are non-dominated in the set of red pairs and blue pairs, respectively. This is true by an obvious exchange argument. Thus, in order to solve the problem it suffices to take each red pair (x,y) and find an optimal partner (x',y') in the sorted sequence U of non-dominated blue pairs. Finally we take the best solution, with maximum $z := \min\{x+y', y+x'\}$.

We distinguish six cases regarding the relationships between the coordinates. In cases of equations, the equality signs $=$ can be arbitrarily replaced with the strict signs $<$ or $>$. (See the Figure.)

(1) $x' < x$ and $y' > y$
(2) $x' < x$ and $y' < y$ and $x - x' > y - y'$
(3) $x' > x$ and $y' > y$ and $x' - x < y' - y$
(4) $x' < x$ and $y' < y$ and $x - x' < y - y'$
(5) $x' > x$ and $y' > y$ and $x' - x > y' - y$
(6) $x' > x$ and $y' < y$

Checking these cases one by one, we see that, if both x' increases y' decreases, then the objective z strictly increases in the regions (1)–(3) and strictly decreases in the regions (4)–(6). Moreover, the sorted sequence U first passes the regions (1)–(3) and then continues in the regions (4)–(6). Hence z is a unimodal discrete function on U, that is, z has only one local maximum which is therefore the global maximum. The maximum can be found by $O(\log h')$ look-ups of function values, by golden section search [10]. Since we have to do this at most h times (for every red pair), the time bound follows. □

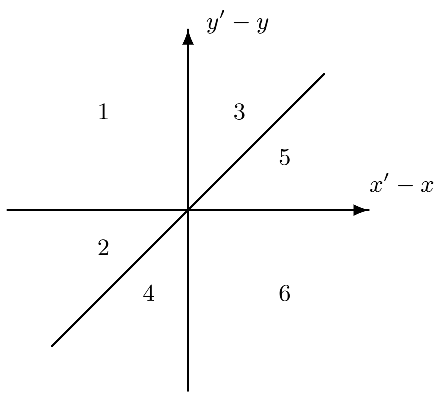

Fig. 1. These are the cases in the proof of Proposition 2.

Remark 1: Due to the search procedures, the log factor in Proposition 2 might be necessary for Largest Mixed Sum in general. Our particular objective function z is actually a "$\leftrightarrow\updownarrow$ unimodal 2D function" in the sense of [7], but we have used unimodality in only one direction. However, it is apparently unknown [7] whether this stronger property allows to find the global maximum in linear time. On another front, we have not established a linear-time reduction from Largest Mixed Sum to the diameter problem. The Largest Mixed Sum instances that can be realized by distances in graphs may have further properties that allow for linear time. We leave these questions open.

6 Diameters Larger than One Third of the Size

Generalizing Lemma 1 we can state, not surprisingly, that graphs with large diameter possess many small separators. We will use this version of the principle:

Lemma 9. *Suppose that $\delta = \frac{1}{2} - h$, where $0 \le h < \frac{1}{6}$. Let P be any longest geodesic path, with r as one of its end vertices. Then at least $(\frac{1}{4} - \frac{3}{2}h)n$ of the layers $N_i(r)$ consist of at most two vertices.*

Proof. The $(\frac{1}{2} - h)n$ layers contain together all n vertices. Define x such that $x + 3(\frac{1}{2} - h - x) = 1$. Then at least xn layers have less than three vertices. Resolving the equation yields the claimed $x = \frac{1}{4} - \frac{3}{2}h$. □

Based on this observation and the result of the previous section we will now propose a randomized algorithm.

Theorem 2. *For every fixed $\delta > 1/3$, a longest geodesic path can be computed with high probability in $O(m + n \log n)$ time.*

Proof. We attempt to construct a longest geodesic path by the following randomized procedure that we call a trial.

Trial, preparation: Choosing separator vertices.

We choose independently three random vertices u, v, w. The following happens with some guaranteed constant probability: (i) Each of u, v, w is in a layer of size at most 2, say $u \in N_i(r)$, $v \in N_j(r)$, $w \in N_k(r)$, where $i < j < k$, moreover, (ii) P goes through u and w. Note that constant probability for (i) holds due to Lemma 9, and for (ii) it follows from (i).

We can replace the subpath from u to w with any geodesic path Q between these vertices (if this geodesic path is not unique), as this yields another geodesic path between the end vertices of P. Thus, without loss of generality we may assume that some particular Q is a subpath of P.

Trial, main phase: Choosing a geodesic path. Observe the following:
(1) If $|N_j(r)| = 1$, then v is an articulation point, moreover, P also goes through v and intersects two different connected components of $G - v$.
(2) If $|N_j(r)| = 2$, then v is not on Q, with constant probability (since both vertices in $N_j(r)$ are proclaimed v with the same probability).

Now we "speculate" that our random u, v, w have properties (i) and (ii) above. Since we do not know which subcase appeared, we proceed as follows.

If v happens to be an articulation point, then situation (1) may be true. In order to capture this possible case, we apply Lemma 2 in order to determine, in $O(n+m)$ time, the longest geodesic path that intersects two different connected components of $G - v$. Since, in particular, P has this property in case (1), we find P or another longest geodesic path in this trial.

If v is not an articulation point, then we know that $|N_j(r)| = 2$, and we speculate that (2) is true. Since Q contains the other vertex of $N_j(r)$, and every layer is a separator S such that P intersects two connected components of $G - S$, we conclude that P also intersects two connected components of $G - (Q \cup \{v\})$. Furthermore, all vertices c and d of P in these two components satisfy $B(c, u, w)$ and $B(u, w, d)$, respectively. Defining the vertex sets $C := \{c | B(c, u, w)\}$ and $D := \{d | B(u, w, d)\}$, we can therefore set up an instance of Largest Mixed Sum, where the numbers x, x', y, y' are the distances of vertices in C and D to

v and to some fixed reference vertex on Q. All these distances are computed by two runs of BFS, with roots u and v, in $O(n+m)$ time.

Largest Mixed Sum returns a path P' with the following properties: P' has its end vertices in C and D, its subpaths in C and D are geodesic, and either P' goes through Q and avoids v, or P' goes through v. (More precisely, only the end vertices of P' are returned, and the information whether P' uses Q or v, but this suffices to finally reconstruct a geodesic path between these ends.) If P' goes through Q (and hence is at most as long as the alternative path through v), we output P' in this trial, otherwise the trial has no output.

Analysis of a trial. In case (1) we have already seen that a longest geodesic path is produced. In case (2), if P actually goes through Q as assumed, then we claim that the trial returns P (or another longest geodesic path). Assume for contradiction that some shorter path P' going through Q is returned. Let us divide P' in three subpaths: (A, Q, B). Since $A \subseteq C$, that is, A contains only vertices c with $B(c, u, w)$, it follows that the subpath (A, Q) is geodesic. By the symmetric argument, (Q, B) is geodesic. Hence, any shortcut on P' must connect A and B jumping over Q, and this is possible only by going through v, since $Q \cup \{v\}$ is a separator. However, by construction the alternative path through v was not shorter, hence P' has no shortcut at all, in other words, P' is geodesic. But since Largest Mixed Sum maximizes the minimum of the two lengths (of the paths through Q and v), it cannot yield a geodesic path shorter than P.

Conclusion. As shown above, our speculative assumptions are true with some guaranteed constant probability, and if they are, the path returned in the trial is in fact a longest geodesic path in the graph. As usual, one can amplify the probability of a correct result to any desired constant close to 1, by repeating the trial $O(1)$ times independently. □

7 Further Research

Does a deterministic algorithm with the same time bound as in Theorem 2 exist? The difficulty is to hit a separator of two vertices that divides some (unknown!) longest geodesic path P. Alternatively we might use a version of Lemma 9 that guarantees a decrease of the largest connected component by a constant factor and thus enables divide-and-conquer, but now the catch is that a separator $S = \{u, v\}$ may have a large $d(u, v)$, and the long subpath $u - v$ of a solution may be in another connected component of $G - S$, such that the size of an instance to be solved recursively does not decrease enough.

The algorithm in Theorem 2 is Monte Carlo. It might be possible to turn it into a Las Vegas algorithm by verifying that the obtained geodesic path P is the longest one. This might be done by a technique as in Theorem 2, but now using the fact that P is already given. (Of course, the question becomes obsolete if a deterministic algorithm can be devised.)

Despite the mentioned difficulties we conjecture that the diameter can be found in nearly linear time for every fixed δ, by some smart use of $O(1/\delta)$ sized separators. By arguments similar to the case $\delta < 1/2$, this would also require a multi-dimensional generalization of Largest Mixed Sum.

Acknowledgment

The author would like to thank the anonymous referees for careful remarks which helped erase a number of small inaccuracies.

References

1. Abboud, A., Grandoni, F., Vassilevska Williams, V.: Subcubic Equivalences Between Graph Centrality Problems, APSP and Diameter. In: Indyk, P. (Ed.) SODA 2015, pp. 1681–1697, ACM-SIAM (2015)
2. Abboud, A., Vassilevska Williams, V., Wang, J.R.: Approximation and Fixed Parameter Subquadratic Algorithms for Radius and Diameter in Sparse Graphs. In: Krauthgamer, R. (Ed.) SODA 2016, pp. 377–391, ACM-SIAM (2016)
3. Boitmanis, K., Freivalds, K., Ledins, P., Opmanis, R.: Fast and Simple Approximation of the Diameter and Radius of a Graph. In: Alvarez, C., Serna, M.J. (Eds.) WEA 2006. LNCS, vol. 4007, pp. 98–108, Springer, Heidelberg (2006)
4. Chechik, Sh., Larkin, D.H., Roditty, L. Schoenebeck, G., Tarjan, R.E., Vassilevska Williams, V.: Better Approximation Algorithms for the Graph Diameter. In: Chekuri, Ch. (Ed.) SODA 2014, pp. 1041–1052, ACM-SIAM (2014)
5. Corneil, D.G., Dragan, F.F., Habib, M., Paul, Ch.: Diameter Determination on Restricted Graph Families. Discr. Appl. Math. 113, 143–166 (2001)
6. Corneil, D.G., Dragan, F.F., Köhler, E.: On the Power of BFS to Determine a Graph's Diameter. Networks 42, 209–222 (2003)
7. Demaine, E.D., Langerman, S.: Optimizing a 2D Function Satisfying Unimodality Properties. In: Brodal, G.S., Leonardi, S. (Eds.) ESA 2005. LNCS, vol. 3669, pp. 887–898, Springer, Heidelberg (2005)
8. Dragan, F.F.: Almost Diameter of a House-hole-free Graph in Linear Time Via LexBFS. Discr. Appl. Math. 95, 223–239 (1999)
9. Hopcroft, J., Tarjan, R.E.: Algorithm 447: Efficient Algorithms for Graph Manipulation. Comm. ACM 16, 372–378 (1973)
10. Kiefer, J.: Sequential Minimax Search for a Maximum. Proc. Amer. Math. Soc. 4, 502–506 (1953)
11. Roditty, L., Vassilevska Williams, V.: Fast Approximation Algorithms for the Diameter and Radius of Sparse Graphs. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (Eds.) STOC 2013, pp. 515–524, ACM (2013)
12. Schmidt, J.M.: A Simple Test on 2-Vertex- and 2-Edge-Connectivity. Info. Proc. Letters. 113, 241–244 (2013)
13. Weimann, O., Yuster, R.: Approximating the Diameter of Planar Graphs in Near Linear Time. ACM Trans. Algor. 12, paper 12 (2016)
14. Yuster, R.: Computing the Diameter Polynomially Faster than APSP. CoRR abs/1011.6181 (2010)