On Farthest-Point Information in Networks*

Prosenjit Bose† Jean-Lou De Carufel† Carsten Grimm†‡ Anil Maheshwari† Michiel Smid†

Abstract

Consider the continuum of points along the edges of a network, an embedded undirected graph with positive edge weights. Distance between these points can be measured as shortest path distance along the edges of the network. We introduce two new concepts to capture farthest-point information in this metric space. The first, eccentricity diagrams, are used to encode the distance towards farthest points for any point on the network compactly. With this, we can solve the minimum eccentricity feed-link problem, i.e., the problem to extend a network by one new point minimizing the largest network distance towards the new point. The second, network farthest-point diagrams, provide an implicit description of the sets of farthest points. A network farthest-point diagram is, in principle, a compressed farthest-point network Voronoi link diagram generated by the entire continuum of uncountably many points on the network at hand. We provide construction algorithms for data structures that allow for queries for the distance to farthest points as well as their location from any point on a network in optimal time. Thus, we establish first bounds on construction times and storage requirements of such data structures.

1 Introduction

The topic of this article was inspired by the following network extension problem introduced by Aronov et. al. [2]. We are given a network of roads and the position of a site, e.g., a hospital, that is not on the network, yet. The site needs to be connected to the existing roads with a new one, referred to as a *feed-link*. Aronov et. al. [2] seek a feed-link that minimizes the largest ratio between the distance to the site via the roads versus the Euclidean distance from any location on the roads. This ratio signifies the largest detour one may take to the site by traveling along the roads as opposed to flying directly to it. When this detour, also referred to as *dilation*, is minimized, the distances via the network resemble the straight line distances as best as possible. An illustration is shown in Figure 1. Grüne [7] provides a summary of dilation and its properties. Notice that all positions on the network are taken into account to evaluate a feed-link and that the feed-link might be connected to any location along the roads. In this sense the dilation is a generalization of the *stretch factor* [11].

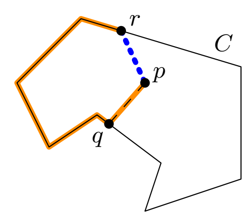


Figure 1: A polygonal cycle C with a point p in it that is connected to C via a feed-link at q. The dilation of point r on the cycle is the ratio between the highlighted path (orange) via the roads to p versus the Euclidean distance between p and r (blue, dotted).

Depending on the application at hand, one might consider other measures. For instance, if the site is a hospital, one might seek to optimize emergency unit response times [5]. Assume an accident occurs along any of the roads, then it is desirable to ensure that the time an emergency crew needs to drive from the hospital to the accident is as small as possible or below a certain critical threshold. Therefore, we seek to minimize the largest road-wise distance to the hospital. The set of farthest locations from the site is precisely the same as that for the meeting point of the feed-link with an existing road. Hence, determining how the set of farthest points and the road-wise distances to them change along the existing roads turns out to be helpful to solve this variant of the feed-link problem.

In this article, we will solve the latter for arbitrary networks of roads using novel data structures that support queries for farthest-point information.

1.1 Problem Definition

A *network* is a straight-line embedding of a simple, finite, connected, and undirected graph $G = (V, E)$, where V is a set of points in $\mathbb{R}^2$, and E is a set of segments whose endpoints are in V. Each edge e has a positive weight $w_e > 0$. A point $p \in \mathbb{R}^2$ is on G, denoted by $p \in G$, if p is on some edge of G. A

---

*Research supported in part by FQRNT and NSERC.

†School of Computer Science, Carleton University, jit@scs.carleton.ca, jdecaruf@cg.scs.carleton.ca, carsten.grimm@ovgu.de, anil@scs.carleton.ca, michiel@scs.carleton.ca

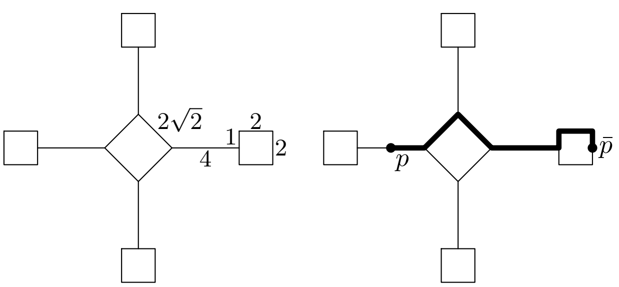
‡Institut für Simulation und Graphik, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg

point p on an edge $uv \in E$ subdivides uv such that $|up| = \lambda\,|uv|$ and $|pv| = (1-\lambda)\,|uv|$ for some $\lambda \in [0,1]$. We choose the weights of the resulting sub-edges up and pv according to the fraction λ, viz., $w_{up} := \lambda w_{uv}$ and $w_{pv} := (1-\lambda)w_{uv}$. A point can only be on one edge. Thus, if a point happens to lie on the proper intersection of two edges, the point can only be associated with one edge. Consider the weighted shortest path distance $d_G\colon V \times V \to [0,\infty)$ between vertices of G with respect to the edge weights w_e, $e \in E$. This can be extended to arbitrary points p and q on G by considering them to be vertices for the sake of evaluating $d_G(p,q)$ [2, 7]. We refer to this as the *network distance* on G. The following definition generalizes a term that is usually introduced with respect to distances between vertices [9, pp. 35–36].

Definition 1 (Eccentricity) *Let G be a network (refer to Figure 2). For a point p on G, the largest network distance towards p is the* eccentricity *of p with respect to G and it is denoted by $\mathrm{ecc}_G(p)$, i.e.,*

$$\mathrm{ecc}_G(p) := \max_{q\in G} d_G(p,q).$$

The point q on G is eccentric *to p if it is a farthest point from p with respect to the network distance, i.e., if $d_G(p,q) = \mathrm{ecc}_G(p)$*[1].



(a) A network G.

(b) A point p on G and a point $\bar{p}$ eccentric to p.

Figure 2: A network G (a) with edge weights as indicated. A point p with its non-vertex eccentric point $\bar{p}$ is shown (b). Here we have $\mathrm{ecc}(p) = 10 + 4\sqrt{2}$ achieved on the highlighted path (black).

Our goal is to design algorithms and data structures for a given network G in order to answer the following types of queries. For any point p on G.

1. What is the eccentricity of p?
2. Which points on G are farthest from p with respect to the network distance?
3. Let uv be an edge such that $p \in uv$. Which points r on uv have the same farthest points as p?

[1]In the remainder of this article we will omit the subscript indicating the underlying network G in all of the above notation when it is clear from the context.

1.2 Related Work

The relation between points p on a network G and their farthest points $\bar{p}$ on G can be expressed in terms of existing notions as follows. It can be stated as the *farthest-point Voronoi diagram* on the metric space $(G, d(\cdot,\cdot))$ where all of the uncountable infinitely many points on G are considered to be *sites* or *generators* of the diagram. Usually *Voronoi diagrams* are computed with respect to a finite set of $n \in \mathbb{N}$ sites. The *farthest-point Voronoi diagram* is a special case of the *k-th nearest neighbor Voronoi diagram* with $k = n$. Even though Voronoi diagrams on networks have been studied before, e.g., [3, 5, 8, 13], they were defined with respect to a finite set of generators. A survey of various notions of Voronoi diagrams, including some for networks, can be found in [12]. Refer to [13] for generalized variants of network Voronoi diagrams and further references.

Information about the eccentricity of points along the edges of a network is also useful in contexts other than the stated feed-link problem. For instance, in the *continuous absolute 1-center problem* from location analysis [4, 14] we seek a point with minimum eccentricity in a network. Furthermore, a point of maximum eccentricity and one of its farthest points form a pair of diametral points. Recent surveys of existing related notions and results can be found in [10, 14].

2 Eccentricity Diagrams

We seek a concise representation of the mapping from the points on a network G to their eccentricity value. Frank [4] seeks a point with minimum eccentricity on G. He finds it by determining the smallest among the minimal eccentricity values on each edge uv. To obtain these values, Frank [4] computes the eccentricity of points on edge uv as a function as follows. Let $\phi^{st}_{uv}\colon [0,1] \to [0,\infty)$ be the mapping such that

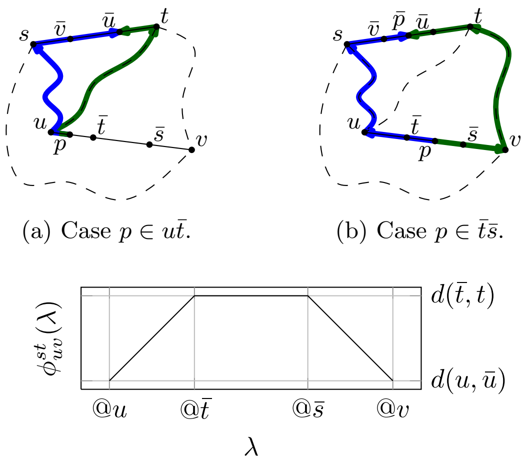
$$[0,1] \ni \lambda \overset{\phi^{st}_{uv}}{\longmapsto} \max_{q\in st} d((1-\lambda)u + \lambda v, q).$$

Consider a point p on edge uv with $p = (1-\lambda)u + \lambda v$, $\lambda \in [0,1]$. The value $\phi^{st}_{uv}(\lambda)$ is the largest network distance from p to any point on edge st. We obtain the eccentricity function for the points p on uv by building the upper envelope of the functions ϕ^{e}_{uv} for all edges e of the network, since

$$\mathrm{ecc}(p) = \max_{q\in G} d(p,q) = \max_{e\in E}\max_{q\in e} d(p,q).$$

The shape of the functions is described in Lemma 2 and depicted in Figure 3.

Lemma 2 ([4]) *Let uv and st be edges of a network G. Then the function ϕ^{st}_{uv} is piece-wise linear with slopes $+w_{uv}$, 0, and $-w_{uv}$ in this order.*



(a) Case $p \in u\bar{t}$. (b) Case $p \in \bar{t}\bar{s}$.

(c) The function ϕ_{uv}^{st}.

Figure 3: The function ϕ_{uv}^{st} for two edges uv an st consist of three linear segments. The point $\bar{u}$ (respectively $\bar{v}$) is the farthest point from u (respectively v) on st. Likewise, the point $\bar{t}$ (respectively $\bar{s}$) is the farthest point from t (respectively s) on uv. Shortest paths attaining the network distance $\phi_{uv}^{st}(p)$ form p to the farthest point $\bar{p}$ from p on st are shown in (a) and (b).

As a consequence, the eccentricity along an edge uv of a network with m edges is the upper envelope of m piece-wise linear functions as in Lemma 2. The domain of all these functions is $[0, 1]$. We can compute this upper envelope using a divide-and-conquer approach described by Agarwal and Sharir [1, Section 2.3].

Lemma 3 *Let uv be an edge of a network G. Any pair of functions ϕ_{uv}^{e} and $\phi_{uv}^{e'}$ for edges e and e' of G intersect at most twice disregarding overlaps.*

Theorem 4 ([1]) *Let $\mathcal{F}$ be a set of k continuous, totally defined functions with a common domain whose graphs intersect in at most two points. The sequence of functions along the upper envelope of $\mathcal{F}$ can be obtained in $\mathcal{O}(k \log(k))$ time and has length at most $2k - 1$.*

With Lemmas 2 and 3 we can use Theorem 4 to estimate the size and construction time of the upper envelope of the functions ϕ_{uv}^{st}.

Corollary 5 *Let uv be an edge of a network G with m edges. The eccentricity on uv is a piece-wise linear and continuous function, consisting of at most $6m - 3$ line segments. It can be computed in $\mathcal{O}(m \log(m))$ time.*

Due to its piece-wise linearity, we can describe the eccentricity completely by stating the value of the eccentricity at the endpoints of each linear segment. That is for any point p in the segment ab with linear eccentricity and with $p = (1 - \lambda)a + \lambda b$, we have

$$\mathrm{ecc}(p) = (1 - \lambda)\,\mathrm{ecc}(a) + \lambda\,\mathrm{ecc}(b).$$

This leads us to the following notion.

Definition 6 (Eccentricity Diagram) *Let G be a network. Consider the subdivisions G' of G with*

$$\mathrm{ecc}(a + \lambda(b - a)) = (1 - \lambda)\,\mathrm{ecc}(a) + \lambda\,\mathrm{ecc}(b),$$

for each edge ab of G' and each $\lambda \in [0, 1]$. Among these we call the one with the least number of vertices the eccentricity diagram *of G and denote it by $\mathcal{ED}(G)$.*

The eccentricity diagram of a network is well-defined and unique, as it can be obtained by subdividing each edge uv at the endpoints of the line segments of the eccentricity function on uv. By Corollary 5, this yields a finite subdivision with the minimum number of additional vertices. An example is shown in Figure 5. As the computation of the upper envelope is performed on each edge, we have the following corollary.

Corollary 7 *The eccentricity diagram of a network with m edges has size $\mathcal{O}(m^2)$ and can be constructed in $\mathcal{O}(m^2 \log(m))$ time, provided the shortest path information between any pair of vertices is known a-priori.*

Next we establish that the size bound stated in Corollary 7 is tight for planar networks. In the full version of this paper we establish a lower bound of $\Omega(nm)$ for general networks with n vertices and m edges.

Lemma 8 *For all $n \in \mathbb{N}$, there exists a (planar) network G with n vertices that has an eccentricity diagram $\mathcal{ED}(G)$ of size $\Omega(n^2)$.*

Proof. Consider the network G depicted in Figure 4 for $k > 2$ and a value of ϵ with $0 < \epsilon < \frac{3}{2(k-2)}$. Each of the k edges $u_i v_i$, $i = 1, \ldots, k$, is subdivided into k sub-edges in the eccentricity diagram of G on $u_i v_i$ by $k - 1$ additional vertices. Thus, we have at least $k(k - 1) \in \Omega(n^2)$ additional vertices in total, as the network has $n = 4k$ vertices. □

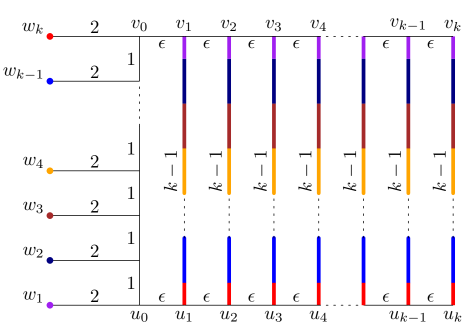
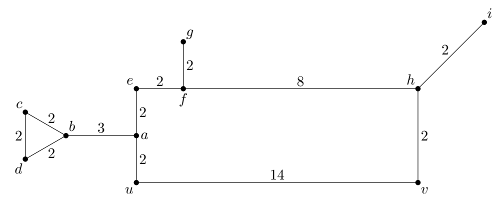
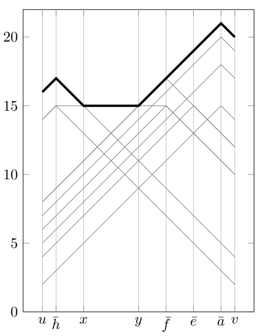


Figure 4: A network whose number of vertices in the eccentricity diagram is quadratic in the number of vertices n in the network itself. Along the edges $u_i v_i$ for $i = 1, \ldots, k$, the farthest point among $w_1, \ldots, w_k$ is indicated in the corresponding colour.



(a) A network.



(b) The functions ϕ_{uv}^{st} and their upper envelope.

Figure 5: An example for the brute-force method applied to the edge uv of the network in (a). The functions ϕ_{uv}^{st} representing the edge-to-edge distances are shown in (b) together with their upper envelope, representing the eccentricity along uv. The vertices $\bar{h}$, x, y, and $\bar{a}$ must be added to uv, to obtain the subdivision of uv in the eccentricity diagram $\mathcal{ED}(G)$ of G.

Assume we are given the eccentricity diagram $\mathcal{ED}(G)$ of a network G as well as the eccentricity values $\mathrm{ecc}(v)$ of all vertices v of $\mathcal{ED}(G)$. Then we can answer queries for the eccentricity value $\mathrm{ecc}(p)$ of a point p on an edge uv using the piecewise linearity of $\mathrm{ecc}(\cdot)$ on uv, where

$$\mathrm{ecc}(p) = \left(1 - \frac{w_{ap}}{w_{ab}}\right) \mathrm{ecc}(a) + \frac{w_{ap}}{w_{ab}}\, \mathrm{ecc}(b),$$

and ab is the sub-edge of uv in $\mathcal{ED}(G)$ containing p. We assume that we are given the edge uv of the original network G containing p when conducting such a query. The sub-edge ab of uv can be found in $\mathcal{O}(\log(n))$ time using binary search as there are at most $6m-3 \in \mathcal{O}(n^2)$ additional vertices on uv in $\mathcal{ED}(G)$ by Corollary 5. The above yields in combination with Corollary 7 and Lemma 8 the following theorem.

Theorem 9 *Given a network G with n vertices and m edges. There is a data structure that can be used to determine the eccentricity value* $\mathrm{ecc}(p)$ *of any point p on G in $\mathcal{O}(\log(n))$ time, provided that the edge uv of G containing p is given. This data structure can be constructed in $\mathcal{O}(m^2 \log(n))$ time, provided that the network distances between all vertices of G are known. The size of this data structure is at most $\mathcal{O}(m^2)$ in general and can be at least $\Omega(n^2)$ for certain planar networks.*

3 Network Farthest-Point Diagrams

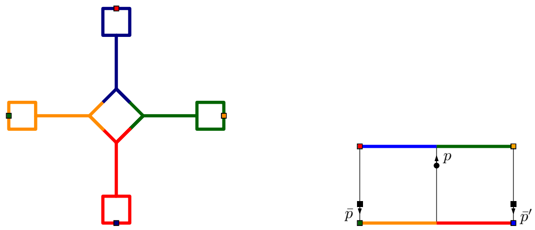
In addition to computing the distance towards farthest points, we are also interested in their location. That is we seek to query for the set of farthest points from any point g on a network G. This suggests the introduction of a continuous version of a farthest-point Voronoi diagram on the metric space formed by the edges of a network and the corresponding network distance.

Definition 10 *Let G be a network. Consider the set*

$$\mathcal{V}_{\mathit{far\text{-}net}}(g) := \{p \in G \colon \forall g' \in G \colon d(p, g') \leq d(p, g)\},$$

of points $p \in G$ whose network distance $d(p, g)$ to a point g on G is largest among the network distances to all other points g' on G. We call $\mathcal{V}_{\mathit{far\text{-}net}}(g)$ the farthest-point network Voronoi link cell *of g. We obtain the* farthest-point network Voronoi link diagram *of G by adding a new vertex to G for each boundary point of the non-empty farthest-point network Voronoi link cells, i.e., at all points of the set $\bigcup_{g \in G} \partial \mathcal{V}_{\mathit{far\text{-}net}}(g)$. If the latter set is finite, we say that the diagram is* finite.

The existing notions of Voronoi diagrams are determined by a finite set of reference points. For instance the *farthest-point Voronoi diagram* [12, Section 3.3] subdivides the plane into regions such that the points in the interior of any region have one common unique farthest point among a finite set of points in $\mathbb{R}^2$. Likewise, the *network Voronoi link diagram* on G [12, Section 3.8] subdivides a network into parts, such that the points in the interior of each part are closest to a common subset of a finite set of points on G. However, for the queries described in Section 1.1, the situation is different. First, we are oblivious of which points g on G are considered farthest points, i.e., satisfy $\mathcal{V}_{\text{far-net}}(g) \neq \emptyset$, when creating the farthest-point network Voronoi link diagram. Thus, the set of reference points is to be determined as opposed to given a-priori. Secondly, this set of reference points may be infinite, as depicted in Figure 6. Therefore, known methods to determine Voronoi diagrams do not necessarily apply here. Moreover, we need to find a way to deal with infinite farthest-point network Voronoi link diagrams. If a finite number of vertices is added, the farthest-point network Voronoi link diagram is a subdivision of G. In that case, it is considered a network itself. Otherwise, it is an *infinite network*, i.e., a network with infinitely many vertices and degenerate edges that may have an empty interior and identical endpoints. In the finite case, it would be sufficient to store the set of farthest points at each vertex and each edge of the farthest-point network Voronoi link diagram to give a full description of the location of eccentric points. However, this is impossible to do explicitly in the infinite case. Next, we will investigate the latter in order to obtain a finite representation of the same information.



(a) Finite diagram. (b) Infinite diagram.

Figure 6: The farthest-point network Voronoi link diagrams for two networks. Parts that have a common farthest point (square) are indicated in colour. In the finite case (a), the network is subdivided into regions with a fixed farthest point. We have a different behaviour on the vertical edges (black) of (b). When the point p is moved upwards, its two farthest points $\bar{p}$ and $\bar{p}'$ move downwards accordingly. No two points on this edge have a common farthest point.

Theorem 11 *Let G be a network. The farthest-point network Voronoi link diagram of G is infinite if and only if there exists an edge ab of the eccentricity diagram $\mathcal{ED}(G)$ of G such that the eccentricity is constant on ab, i.e., we have* $\mathrm{ecc}(a) = \mathrm{ecc}(p)$ *for all $p \in ab$.*

We distinguish two types of phenomena on the edges of the eccentricity diagram. On edges uv with non-constant eccentricity, the farthest points are stationary in the sense that uv can be subdivided into finitely many sub-edges without any change of the farthest-point set in their interior. On edges with constant eccentricity however, each point has its own set of farthest points distinct from that of any of the uncountably many other points on it. Nonetheless, we can subdivide edges that exhibit the latter behavior into finitely many portions, such that the farthest points on each portion are contained in a common set of edges. This simplification yields a finite representation of the farthest-point network Voronoi link diagram defined as follows.

Definition 12 (Farthest-Point Diagram) *Let G be a network. Consider the subdivisions G' of the eccentricity diagram $\mathcal{ED}(G)$ of G such that each edge uv of G' is of one of the following types.*

(i) The eccentricity on uv is non-constant, and all points in the interior of uv have the same set of farthest points in G.

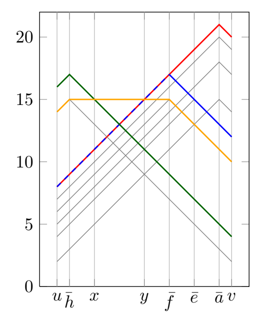
(ii) The eccentricity on uv is constant, and all points in the interior of uv have the same set of edges of G containing their farthest points in G.

Among these subdivisions we call the one with the least number of additional vertices the network farthest-point diagram *of G and denote it by $\mathcal{FD}(G)$.*

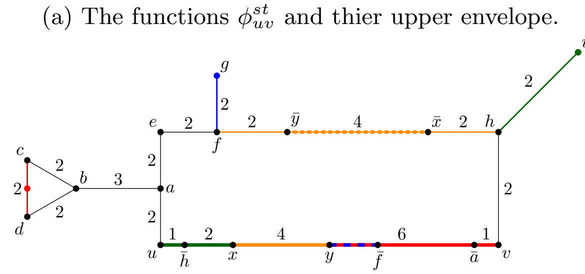
The network farthest-point diagram can be obtained in the same manner as the eccentricity diagram. During the construction of the upper envelope, we keep track of the edges e whose functions ϕ^e_{uv} contribute to this envelope. See Figure 7 for an example.

For each edge of the network farthest-point diagram we can store the set of farthest points or the set of edges containing them depending on the type of the edge. Each of these sets consist of at most m elements. With this data we can answer queries for the set of farthest points of a point p on a given edge uv of G as follows. First, we identify the sub-edge ab of uv in $\mathcal{FD}(G)$ containing p using binary search. If a set of farthest points is stored with ab, we return this set. Otherwise, we store the set of edges containing the farthest points of p with ab. In that case we use the distances between all vertices of G to obtain the locations of the farthest points from p in constant time per point.

Theorem 13 [2] *Given a network G with n vertices and m edges. There is a data structure that can be used to determine the set of farthest points of any point p on G in $\mathcal{O}(\log(n)+k)$ time, when given the edge uv containing p, where k is the size of the output. This data structure has a construction time and size of $\mathcal{O}(m^3)$.*



(a) The functions ϕ^{st}_{uv} and thier upper envelope.



(b) The network farthest-point diagram on uv indicated with colours. Farthest points are located at the dot(ted segments) of matching colour.

Figure 7: An example for determining the network farthest-point diagram for the network G from Figure 5. The upper envelope (a) of the functions ϕ^{st}_{uv} reveals which edges contain farthest points (b).

[2]In the full version of this paper, we show how to obtain this result with a construction time and size bound of $\mathcal{O}(m^2 \log(n))$.

4 Solving a Feed-Link Problem

Now we demonstrate how to solve the feed-link problem stated in the introduction with the aid of the data structure from Theorem 9. We begin with a formal definition of the former. Here we assume all edge weights, including that of any possible feed-link, to be equal to the Euclidean length of the corresponding line segment. A network with this property is referred to as *geometric*. Further, if we introduce the feed-link pq to a point q on G, we denote the resulting network by $G + pq$ and refer to q as the *anchor* of p in $G + pq$.

Definition 14 *Let G be a geometric network. Further, let p be a point in the plane that is not on G. We call the problem of determining a point q on G such that the eccentricity of p with respect to $G + pq$ is smallest the* minimum eccentricity feed-link problem*.*

Lemma 15 *Let uv be an edge of the eccentricity diagram of G. If the eccentricity is increasing on uv from u to v, then u is the optimal anchor on uv. Otherwise, the closest point to p on uv is the optimal anchor on uv.*

The (globally) optimal anchor on G is found by scanning through all edges of the eccentricity diagram of G and determining the (locally) optimal anchor on each of them. In case there are restrictions for the position of the anchor point, we only use the part of the eccentricity diagram for the allowed anchor points. For example, one could require that the extended network $G + pq$ should be planar. Then only the points on G that are visible from p may be anchors.

5 Future Work

The construction algorithms for the data structures in Theorem 9 and 13 work for any type of network, yet they suffer from slow running times and the need to know all vertex-to-vertex distances in the network. The following improvements [6] upon these results are beyond the scope of this extended abstract and will be the matter of future publications. For cactus networks, we can obtain data structures with the same query times as in Theorem 9 and 13 but with storage requirement and construction time of $\mathcal{O}(n)$. Moreover, for planar networks, we can construct a data structure for a designated face in $\mathcal{O}(n \log(n))$ time. Neither of these results require precomputed vertex-to-vertex distances. For more details refer to Grimm [6].

References

[1] P. K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. In *Handbook of computational geometry*, pages 1–47. North-Holland, 2000.

[2] B. Aronov, K. Buchin, M. Buchin, B. M. P. Jansen, T. de Jong, M. J. van Kreveld, M. Löffler, J. Luo, R. I. Silveira, and B. Speckmann. Connect the dot: Computing feed-links for network extension. *Journal of Spatial Information Science*, 3(1):3–31, 2011.

[3] M. Erwig. The graph Voronoi diagram with applications. *Networks*, 36(3):156–163, 2000.

[4] H. Frank. A note on a graph theoretic game of Hakimi's. *Operations Research*, 15(3):567–570, 1967.

[5] T. Furuta, A. Suzuki, and K. Inakawa. The k-th nearest network Voronoi diagram and its application to districting problem of ambulance systems, 2005.

[6] C. Grimm. Eccentricity diagrams. Diplomarbeit (Master's thesis), Otto-von-Guericke-Universität Magdeburg, Magdeburg, 2012.

[7] A. Grüne. *Geometric Dilation and Halving Distance*. Dissertation (PhD thesis), Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, 2006.

[8] S. L. Hakimi, M. Labbé, and E. F. Schmeichel. The Voronoi partition of a network and its implications in location theory. *INFORMS Journal on Computing*, 4(4):412–417, 1992.

[9] F. Harary. *Graph theory*. Narosa/Addison-Wesley, indian student edition edition, 1989.

[10] R. K. Kincaid. Exploiting structure: Location problems on trees and treelike graphs. In *Foundations of Location Analysis*, volume 155 of *International Series in Operations Research & Management Science*, pages 315–334. Springer US, 2011.

[11] G. Narasimhan and M. Smid. *Geometric spanner networks*. Cambridge University Press, 2007.

[12] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*. Wiley Series in Probability and Statistics. John Wiley & Sons Ltd., Chichester, 2nd edition, 2000.

[13] A. Okabe, T. Satoh, T. Furuta, A. Suzuki, and K. Okano. Generalized network Voronoi diagrams: Concepts, computational methods, and applications. *International Journal of Geographical Information Science*, 22(9):965–994, 2008.

[14] B. Ç. Tansel. Discrete center problems. In *Foundations of Location Analysis*, volume 155 of *International Series in Operations Research & Management Science*, pages 79–106. Springer US, 2011.