Geometric Ad-Hoc Routing for Unit Disk Graphs and General Cost Models

Report

Author(s):
Kuhn, Fabian; Wattenhofer, Roger; Zollinger, Aaron

Publication date:
2002

Permanent link:
https://doi.org/10.3929/ethz-a-006654569

Originally published in:
Technical Report / ETH Zurich, Department of Computer Science 373

Geometric Ad-Hoc Routing for Unit Disk Graphs and General Cost Models

Fabian Kuhn, Roger Wattenhofer, Aaron Zollinger
{kuhn,wattenhofer,zollinger}@inf.ethz.ch
Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland

Abstract

What is the influence of the chosen cost metric on the performance of a mobile ad-hoc routing algorithm? In this paper we define the notion of a general cost metric and observe that all cost metrics fall into two classes, linearly bounded and super-linear. Distinguished by a natural argument, the two classes yet show a dramatic difference: On a network with linearly bounded cost metric a geometric routing algorithm will find a route whose cost at most quadratic in the cost of the optimal route, which at the same time is asymptotically optimal. On the other hand there is no such bound on a graph with super-linear cost functions for any geometric routing algorithm. We introduce, however, the class of bounded degree unit disk graphs, on which all cost metrics are equivalent. We finally propose an asymptotically optimal distributed geometric routing algorithm based on node clustering and network backbone construction.

1 Introduction

One of the crucial points in the analysis of mobile ad-hoc routing algorithms is the chosen path cost model. What weight do we assign to the edges in the network graph? Should the cost of a route represent the number of intermediate network nodes? Should it rather describe the Euclidean length of the route? Is it more suitable to consider transmission energy? Or is yet another measure or even a combination of different metrics reasonable? In summary: What is possible with which metric? In this paper we analyse such cost models for mobile ad-hoc networks. Particularly we propose a definition of a general cost model and show that these metrics naturally fall into two classes which differ dramatically with respect to performance bounds of routing algorithms.

In mobile ad-hoc networks nodes communicate directly via wireless radio without stationary infrastructure. Two nodes hear each other if their distance is not greater than a transmission range R. When scaling the transmission range to 1, we obtain a *unit disk graph* describing the network.

If the source and the destination of a message cannot communicate directly, the message can be forwarded by intermediate nodes. This process is referred to as routing in wireless ad-hoc networks. In particular we consider geometric routing, which assumes that (1) each network node knows its own position and the positions of its neighbors and (2) the source sending a message knows the position of the message destination.

For our general cost model we allow any nondecreasing cost function defined in the interval $]0, 1]$. We show that these functions are naturally partitioned into two classes, *linearly bounded* and *super-linear*. The cost metric functions are classified according to their behavior for edges approaching zero length. Informally speaking, all cost functions approximating zero faster then the edge length belong to the second class. Of the probably most thoroughly studied metrics—link (hop) distance, Euclidean distance, and energy cost—the first two are linearly bounded, whereas the energy metric is super-linear.

Astonishingly, a super-linear cost metric allows to construct a chain of very close nodes over which it is possible to cover a finite Euclidean distance for "free", i.e. for zero costs in the limit. We show that using a super-linear edge cost function there are graphs containing this construction on which no geometric routing algorithm can compute a route whose cost is bounded by the cost of the optimal route. On the other hand, all linearly bounded cost functions are equivalent in the sense that there are geometric algorithms which find a route whose cost is upper-bounded by $c^2(p^*)$, where p^* is the optimal route with respect to the cost $c(\cdot)$. At the same time this is shown to be a tight upper bound, i.e. there are graphs on which no geometric routing algorithm can reach a better result.

We also propose an algorithm which does compute (asymptotically) optimal results for linearly bounded cost metrics. Initially this algorithm constructs a backbone of the network. This backbone is a *bounded degree unit disk graph*. We show that for this class of graphs interestingly the above discrimination of two cost metric classes does not hold anymore: On bounded degree unit disk graphs our algorithm finds routes quadratic in the cost of the optimal route also for super-linear cost functions. After forming the network backbone, actual routing takes place by means of the geometric routing algorithm AFR introduced in [11].

The paper is organized as follows: Section 2 summarizes relevant previous work, Section 3 introduces the model and notation we use, and Section 4 defines the cost model. Sections 5 and 6 analyze routing on bounded degree unit disk graphs and on general unit disk graphs, respectively. The paper is concluded in Section 7.

2 Related Work

Geometric routing for mobile ad-hoc networks has been proposed already over a decade ago [4, 6, 15]. In geometric routing each network node knows its own position and the positions of its neighbors. Additionally, the source of a message knows the position of the destination. Above all the first of these conditions has become more and more realistic with the advent of inexpensive positioning systems, such as GPS or Galileo, particularly by equipping each network node with an according receiver [7] or by local information exchange with neighboring nodes [14]. But also the availability of information about the position of the destination is conceivable by means of a separate location service, such as an overlay peer-to-peer network maintaining destination positions [12]. In another setting—usually referred to as geocasting [8, 13]—it might be desirable to send a message to any network node in a certain area.

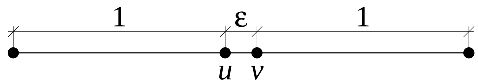

Figure 1: In order to maintain connectivity, both u and v have to be cluster heads, even for arbitrarily small ε.

The early proposals of geometric routing were of greedy nature. However, neither routing of the message to the neighbor closest to the destination [4, 6, 15] nor a "least deviation angle" approach called *Compass Routing* [9] guarantee message delivery in all cases. The first geometric routing algorithm that does guarantee delivery was *Face Routing* introduced in [9] (called *Compass Routing II* there). Face Routing reaches the destination after $O(n)$ steps, n being the number of network nodes. There have been later suggestions for algorithms with guaranteed message delivery; at least in the worst case, however, none of them outperforms original Face Routing. Yet other geometric routing algorithms have been shown to reach the destination on special planar graphs without any runtime guarantees. A more detailed overview of geometric routing can be found in [16].

In [11] we proposed *Adaptive Face Routing* AFR. With this algorithm based on Face Routing it is possible to bound the cost of the computed route by the cost of the optimal route. In particular, the cost of the route found by AFR is not greater than the squared cost of the optimal route. We also showed that this is the optimal result any geometric routing algorithm can reach.

The analysis of AFR is based on the $\Omega(1)$-model, the assumption that the distance between any pair of network nodes is greater than a constant. A straightforward approach to computing such a graph from an arbitrary given network could be sought in clustering, i.e. by grouping nodes around cluster heads, such that finally the graph induced by the cluster heads is an $\Omega(1)$-graph. A counterexample however shows that this approach fails (see Figure 1).

Using results proposed in [17], it is nevertheless possible to construct a graph with bounded degree featuring all desired properties for our purposes. Clustering for the means of ad-hoc routing has been proposed by various researchers [3, 10]. A closely related approach is the construction of *connected* *dominating sets* as routing backbones [5, 18].

3 Model

The purpose of this section is to define the model and the notation we use. The focus of this paper lies on geometric ad-hoc routing algorithms particularly on the unit disk graph. More specifically, we assume that all nodes are placed in the Euclidean plane $\mathbb{R}^2$. Hence, the communication graph is a Euclidean graph, i.e. a weighted graph whose edge weights correspond to the Euclidean distances between adjacent nodes. A graph $G := (V, E)$ is defined by its node set V and its edge set $E \subseteq V^2$. $n := |V|$ denotes the number of nodes, and the Euclidean length of an edge $e \in E$ is denoted by $d(e)$. A path $p := v_1, \ldots, v_k$ for $v_i \in V$ is a list of adjacent nodes in G, i.e. $(v_i, v_{i+1}) \in E$. Note that a node can occur multiple times when going along a path p, i.e. p may contain cycles. Alternatively, we also denote a path p by the corresponding list of edges.

As already mentioned, we consider the widely used model where all nodes have the same transmission range R. This means that the neighborhood of each node consists of all nodes with distance at most R and consequently that all links are bidirectional. For convenience (and without loss of generality), we scale the distances such that the transmission range become 1. The resulting graph is a unit disk graph (UDG).

Definition 3.1. (Unit Disk Graph) *Let $V \subset \mathbb{R}^2$ be the set of nodes and let $E \subseteq V^2$ be the set of edges such that $(u, v) \in E$ if and only if the Euclidean distance between u and v is at most 1. Then the Euclidean graph $G = (V, E)$ is called the unit disk graph of the nodes in V.*

We now give the definition of a geometric ad-hoc routing algorithm.

Definition 3.2. (Geometric Ad-Hoc Routing Algorithm) *Let $G = (V, E)$ be a Euclidean graph. The purpose of a geometric ad-hoc routing algorithm $\mathcal{A}$ is to transmit a message from a source $s \in V$ to a destination $t \in V$ by sending packets over the edges of G while complying with the following conditions:*

- *Initially all nodes $v \in V$ know their geometric positions as well as the geometric position of all of their neighbors in G.*
- *The source s knows the position of the destination t.*
- *A node is not allowed to store anything except for local information and temporarily stored packets in transit.*
- *The additional information which can be stored in a packet is limited by $\mathrm{O}(\log n)$ bits, i.e. information about $\mathrm{O}(1)$ nodes is allowed.*

In the past geometric ad-hoc routing has been given various names, such as O(1)-memory routing algorithm in [1, 2], local routing algorithm in [9], or position-based routing. Due to the storage restrictions, geometric ad-hoc routing algorithms are inherently local.

4 Cost Model

To measure the quality of a routing algorithm, we give each edge e a cost which is a function of the Euclidean length of e.

Definition 4.1. (Cost Function) *A cost function $c\colon\,]0,1] \mapsto \mathbb{R}^+$ is a nondecreasing function which maps any possible edge length d $(0 < d \leq 1)$ to a positive real value $c(d)$ such that $d' > d \Longrightarrow c(d') \geq c(d)$. For the cost of an edge $e \in E$ we also use the shorter form $c(e) := c(d(e))$.*

Note that $]0, 1]$ really is the domain of a cost function $c(\cdot)$, i.e. $c(\cdot)$ has to be defined for all values in this interval and in particular, $c(1) < \infty$. The cost model defined by such cost functions includes all popular cost measures such as the link distance metric ($c(d) \equiv 1$), the Euclidean distance metric ($c(d) := d$), energy ($c(d) := d^\alpha$ for $\alpha \geq 2$), as well as hybrid measures which are positive linear combinations of above metrics. Although from a theoretical point of view a more general notion of cost functions would be possible, it does not appear to be reasonable to consider other (i.e. not nondecreasing) functions.

For convenience we also define the cost of paths and algorithms. The cost of a path $p = e_1, \ldots, e_k$ is defined as the sum of the costs of its edges:

$$c(p) := \sum_{i=1}^{k} c(e_i).$$

Analogously, the cost $c(\mathcal{A})$ of an algorithm $\mathcal{A}$ is defined as the sum of the costs of all edges which are traversed during the execution of an algorithm on a particular graph.

Sometimes the model allows to simultaneously send a packet to more than one direct neighbor by sending a single message. In this case the cost of sending a packet to a group of neighbors with maximal distance $d_{\max}$ is $c(d_{\max})$. We do not use this for our algorithm. With our lowerbounds, we can even show that it has no asymptotic effect.

Because all our results remain unchanged if simultaneous sending to multiple neighbors is allowed, we will not consider this case any further.

As Section 6 will show, the behavior around zero divides the cost functions into two natural classes. The cost functions which are lower-bounded by a linear function are called **linearly bounded cost functions**, the cost functions which are not bounded by a linear function are called **super-linear cost functions**:

$$\text{linearly bounded} \quad : \quad \lim_{d \to 0} \frac{c(d)}{d} > 0,$$
$$\text{super-linear} \quad : \quad \lim_{d \to 0} \frac{c(d)}{d} = 0.$$

Note that because $c(d)$ is nondecreasing and since it has to be defined for all $d \in]0, 1]$, it can be shown that the above limit exists (∞ allowed).

5 Bounded Degree Unit Disk Graphs

In [11] we show how to do geometric ad-hoc routing on unit disk graphs with a minimum distance d_0 between any two nodes (we called this the $\Omega(1)$-model). In this section we generalize this model to unit disk graphs where the degree of each node is upper-bounded and show that *Adaptive Face Routing* as introduced in [11] still is asymptotically optimal for all possible cost functions (cf. Definition 4.1).

Definition 5.1. (Bounded Degree Unit Disk Graph) *Let G be the unit disk graph for a given set $V \in \mathbb{R}^2$ of points in the plane. If the degree of each node in G is bounded by a predefined constant k, G is called a bounded degree unit disk graph with parameter k.*

Note that if the distance between any two points is lower-bounded by a constant d_0 ($\Omega(1)$-model), the resulting unit disk graph is a bounded degree unit disk graph with $k \le 4/d_0^2 + 4/d_0 + 1$, because the disks with radius $d_0/2$ around each node are disjoint and all the disks of the neighbors of a node u have to be completely inside the disk with radius $1 + d_0/2$ around u.

For the analysis of AFR on bounded degree unit disk graphs, it will be important to know about the number of nodes in a given two-dimensional region. This leads to the next lemma.

Lemma 5.1. *Let $R \subset \mathbb{R}^2$ be a (closed) two-dimensional convex region with area $A(R)$ and perimeter $p(R)$. Further, let $V \subset R$ be a set of points inside R. If the unit disk graph of V is a bounded degree unit disk graph with parameter k (all degrees are at most k), the number of points in V is bounded by*

$$|V| \le (k+1)\frac{8}{\pi}\left(A(R) + p(R) + \pi\right).$$

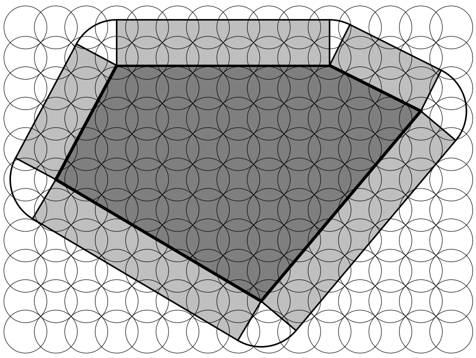

Figure 2: Covering a convex region with a grid of equally sized disks

Proof. In order to prove Lemma 5.1, we first consider the disks with diameter 1. All nodes inside such a disk are less than 1 apart and are therefore adjacent in the unit disk graph. Since the number of neighbors of each node is bounded by k, each disk with diameter 1 contains at most $k+1$ nodes. In order to give a bound on the number of nodes inside the region R, we therefore have to find an upper bound on the number of disks with diameter 1 needed to completely cover R. We can cover the whole plane with disks of diameter 1 by placing the disks on an orthogonal grid such that the horizontal and the vertical distances between the centers of two neighboring disks are $1/\sqrt{2}$ (see Figure 2). By counting the number of disks intersecting R, we get a bound on the number of disks needed to cover R. We see that all disks intersecting R are completely inside the region R', where R' is defined as the locus of all points whose distances from R are at most 1, i.e. we add a border of width 1 to R. Let A' be the area covered by R'. The number of disjoint disks with diameter 1 which can be placed inside R' is bounded by $4A'/\pi$ (the area of a disk with diameter 1 is $\pi/4$) and since in the above defined grid of disks no point in $\mathbb{R}^2$ is covered by more than 2 disks, the number of disks needed to cover R can be bounded by $8A'/\pi$. Thus, the number of nodes in V is at most $(k+1)8A'/\pi$.

In order to get the area A', it is sufficient to consider the case where R is a convex polygon. The general case then follows by limit considerations. We get A' by adding $A(R)$ (the area of R) and the area of the border around R. As illustrated in Figure 2, the border can be composed into rectangles and sectors of circles. For each side of the polygon R we get a rectangle of width 1, and since all the angles of the sectors add up to 2π, the sectors add up to a disk of radius 1. For A' we therefore get $A' = A(R) + p(R) + \pi$ where $p(R)$ denotes the perimeter of R. This concludes the proof. $\square$

We could get a better constant than $8/\pi$ by taking a hexagonal grid and considering the portion of the plane which is only covered by a single disk.

The next lemma shows that up to a constant factor all metrics defined by cost functions are equivalent on bounded degree unit disk graphs.

Lemma 5.2. *Let $c_1(\cdot)$ and $c_2(\cdot)$ be cost functions as defined in Definition 4.1 and let G be a bounded degree unit disk graph with node set V and maximum node degree k. Further let p be a path from $s \in V$ to $t \in V$ on G such that no node occurs more than once in p, i.e. p is cycle-free. We then have*

$$c_1(p) \leq \alpha c_2(p) + \beta$$

for two constants α and β, i.e. $c_1(p) \in \Theta(c_2(p))$.

Proof. Let $c_d(x) := x$ be the cost function of the Euclidean distance metric. We show that for any cost function c there exist constants α_1, β_1, α_2, and β_2 such that

$$c(p) \leq \alpha_1 c_d(p) + \beta_1 \text{ and} \tag{1}$$

$$c(p) \geq \alpha_2 c_d(p) + \beta_2. \tag{2}$$

This means that all cost functions are in $\Theta(c_d(p))$ and particularly $c_1(p) \in \Theta(c_d(p))$ and $c_2(p) \in \Theta(c_d(p))$ which proves the lemma.

We start with Inequality (1). Let $c_\ell(x) \equiv 1$ be the cost function of the link distance metric. Now pick a node u from the path p. Because u has at most k neighbors, we leave the disk with radius 1 around u after at most $k+1$ steps when starting at u and walking along p. Therefore, the total Euclidean distance of any $k+1$ subsequent edges of p is at least 1. We then have

$$c_\ell(p) < (k+1)\lceil c_d(p)\rceil \leq (k+1)c_d(p) + k + 1.$$

Because cost functions are monotone increasing, we have $c(e) \leq c(1)$ for any edge e and any cost function $c(\cdot)$. Therefore, we get

$$c(p) < c(1) \cdot c_\ell(p) \leq (k+1)c(1)\left(c_d(p) + 1\right),$$

which proves Inequality (1). Note that as soon as the cost function $c(\cdot)$ is fixed, $c(1)$ is a constant since we required $c(x)$ to be defined for all $x \in]0,1]$. In order to obtain Inequality (2), we observe that a path p' of length $c_d(p') \geq 1$ has at least one edge e' of length $c_d(e') \geq 1/(k+1)$. If p' consists of $m < k+1$ edges, the longest edge of p' has at least length $1/m$; if p' consists of $k+1$ or more edges, we use the fact that $k+1$ subsequent edges of p have a total Euclidean length of at least 1. We now partition p into maximal subpaths of length smaller than 2. All but the last of these subpaths have a Euclidean length which is at least 1 and therefore we have

$$\begin{aligned} c(p) &\geq c\left(\frac{1}{k+1}\right) \cdot \lfloor \frac{c_d(p)}{2} \rfloor \\ &> c\left(\frac{1}{k+1}\right) \cdot \left(\frac{c_d(p)}{2} - 1\right), \end{aligned}$$

which concludes the proof. $\square$

As an application of Lemma 5.2 we get the following lemma.

Lemma 5.3. *Let G be a bounded degree unit disk graph with node set V. Further let $s \in V$ and $t \in V$ be two nodes and let p_1^* and p_2^* be optimal paths from s to t on G with respect to the metrics induced by the cost functions $c_1(\cdot)$ and $c_2(\cdot)$, respectively. We then have*

$$c_1(p_2^*) \in \Theta(c_1(p_1^*)) \text{ and } c_2(p_1^*) \in \Theta(c_2(p_2^*)),$$

i.e. the costs of optimal paths for different metrics only differ by a constant factor.

Proof. By the optimality of p_2^*, we get

$$c_2(p_1^*) \geq c_2(p_2^*). \quad (3)$$

p_1^* and p_2^* are cycle free and therefore we can apply Lemma 5.2. We then get

$$c_2(p_1^*) \in \Theta(c_1(p_1^*)) \text{ and } c_1(p_2^*) \in \Theta(c_2(p_2^*)). \quad (4)$$

Combining Equations (3) and (4) yields $c_1(p_2^*) \in \mathrm{O}(c_1(p_1^*))$. But by the optimality of p_1^* we have $c_1(p_2^*) \geq c_1(p_1^*)$ and therefore, $c_1(p_2^*) \in \Theta(c_1(p_1^*))$ holds. The second equation of the lemma then follows by symmetry. □

Our routing algorithm AFR works on planar graphs, we therefore have to find a suitable planar subgraph of the unit disk graph. Note that we always use the term planar graph for Euclidean planar graphs, i.e. we consider an embedding in the plane. There are several standard approaches to obtain a planar subgraph of the unit disk graph, one of which is the Gabriel Graph (GG). We will now show that the Gabriel Graph has all required properties. The Gabriel Graph of a set $V \in \mathbb{R}^2$ of nodes is defined as the set of all edges $(u, v) \in V^2$ such that no other point $w \in V$ is inside or on the circle which has $\overline{uv}$ as a diameter. It is well known that the intersection between the Gabriel Graph and the unit disk graph (GG ∩ UDG) is connected iff the UDG is connected. It is also well known that GG ∩ UDG contains an energy optimal path (see Figure 7 in [11]). This leads to the next lemma.

Lemma 5.4. *Let G be a bounded degree unit disk graph with node set V and let G_{GG} be the intersection of G and the Gabriel Graph of V. Further, we fix two nodes $s \in V$ and $t \in V$. Let $c(\cdot)$ be a cost function and p^* and p^*_{GG} be optimal paths with respect to the metric $c(\cdot)$ on G and on G_{GG}, respectively. We then have*

$$c(p^*_{\mathrm{GG}}) \in \Theta(c(p^*)),$$

i.e. G_{GG} is a spanner for all cost functions.

Proof. As already mentioned, it is well known that G_{GG} contains an optimal path with respect to the metric corresponding to the cost function $c(d) := d^2$ (in fact, this also holds for exponents $\alpha > 2$). By applying Lemma 5.3, we now see that the optimal energy path p^*_{E} is competitive for all cost functions $c(\cdot)$, i.e. $c(p^*_{\mathrm{E}}) \in \Theta(c(p^*))$. □

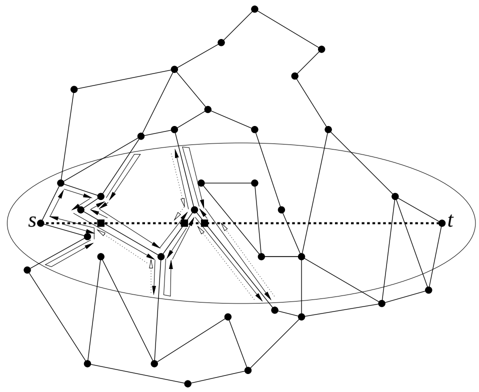

Figure 3: *Bounded Face Routing* (no success: $\widehat{c_d}$ is chosen too small)

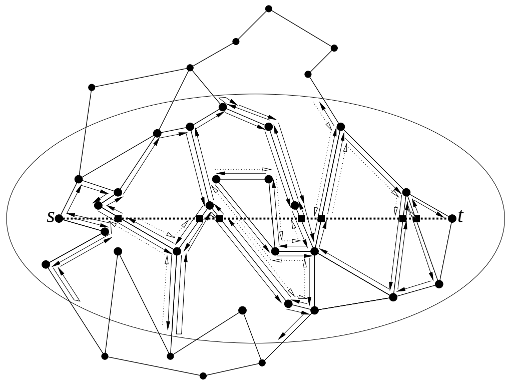

Figure 4: Successful *Bounded Face Routing*

In the following, we describe the core component of our routing algorithm. For completeness, we briefly review *Adaptive Face Routing* (AFR), a geometric ad-hoc routing algorithm, which we introduced in [11]. The key ingredient of AFR is *Bounded Face Routing* (BFR), which itself is an adaptation of *Face Routing* introduced in [9]. The idea of BFR is to perform *Face Routing* while staying inside a finite region containing the source and the destination. For a full description see [11].

Bounded Face Routing (BFR[$\widehat{\mathbf{c_d}}$]): Let E be the ellipse which is defined by the locus of all points the sum of whose distances from s and t is $\widehat{c_d}$, i.e. E is an ellipse with foci s and t.

0. Start at s and let F be the face which is incident to s and which is intersected by $\overline{st}$ in the

immediate region of s.

1. We explore the face F and remember the intersection point p of $\overline{st}$ with the edges of F which is nearest to t. We start the exploration by walking into one of the two possible directions. We continue until we come around the whole face F (as in the normal *Face Routing* algorithm) or until we would cross the boundary of E. In the latter case, we turn around and walk in the opposite direction until we hit the boundary of E again. If the exploration of F does not give a better p (we find the same p as in the last iteration), *Bounded Face Routing* does not find a route to t and we restart BFR to find a route back from p to the source s by using the same ellipse E. Otherwise, proceed with step 2.

2. p divides $\overline{st}$ into two line segments where $\overline{pt}$ is the not yet "traversed" part of $\overline{st}$. Update F to be the face which is incident to p and which is intersected by the line segment $\overline{pt}$ in the immediate region of p. Go back to step 1.

Figure 3 shows an example where BFR does not find a path from s to t, because the ellipse, i.e. the parameter $\widehat{c_d}$, is chosen too small. Figure 4 shows a successful execution of the *Bounded Face Routing* algorithm. To obtain the final routing algorithm AFR, we apply BFR iteratively with exponentially growing $\widehat{c_d}$ until we succeed and finally arrive at t.

Adaptive Face Routing (AFR):

0. Set the size of the ellipse for the first iteration by initializing $\widetilde{c_d}$, e.g. $\widetilde{c_d} := 2\overline{st}$.

1. Execute BFR[$\widetilde{c_d}$].

2. If the BFR execution of step 1 succeeded, we are done; otherwise, we double the estimate for the length of the shortest path ($\widetilde{c_d} := 2\widetilde{c_d}$) and go back to step 1.

We are now able to apply AFR on bounded degree unit disk graphs:

Theorem 5.5. *Let G be a bounded degree unit disk graph and let G_{GG} be the intersection of the corresponding Gabriel Graph and G. Let p^* be a shortest path from the source s to the destination t on G with respect to the metric defined by the cost function $c(\cdot)$. The cost of AFR when applying it on G_{GG} to find a route from s to t then is quadratic in $c(p^*)$:*

$$c(\mathrm{AFR}) = \mathrm{O}\big(c^2(p^*)\big)\,.$$

Proof. In [11] we showed that the link distance cost $c_\ell(\cdot)$ of *Bounded Face Routing* is linear in the number of nodes inside the ellipse E, no matter if BFR finds a path from s to t or if it has to turn around and go back to s. To bound the number of nodes inside ellipse E, we apply Lemma 5.1. As before, we assume that $\widehat{c_d}$ denotes the sum of the distances of a point on the boundary of E from the two foci of E. For the perimeter $p(E)$ and the area $A(E)$ of the ellipse we then have $p(E) \in \mathrm{O}(\widehat{c_d})$ and $A(E) \in \mathrm{O}(\widehat{c_d}^2)$. Because E is a convex region, we can therefore conclude that the number of nodes of G inside E is bounded by $\mathrm{O}(\widehat{c_d}^2)$. Therefore we get:

$$c_\ell(\mathrm{BFR}) = \mathrm{O}\left(\widehat{c_d}^2\right). \tag{5}$$

From [11], we also know that BFR succeeds if and only if there is a path from s to t on G which completely lies inside (or on) the ellipse E. Let p_d^* be an optimal path (on G) with respect to the Euclidean distance metric. By the definition of E, p_d^* lies inside E if $\widehat{c_d} \geq c_d(p_d^*)$, where $c_d(\cdot)$ denotes the cost function for the Euclidean distance. For AFR this means that step 1 succeeds as soon as $\widetilde{c_d} \geq c_d(p_d^*)$ at the latest. The maximum $\widetilde{c_d}$ is at most $2c_d(p_d^*)$, and therefore the cost of the last application of BFR is $\mathrm{O}(c_d^2(p^*))$. The sum of the costs of all iterations (i.e. applications of BFR) is a geometric series and therefore linear in the cost of the greatest summand of the series, i.e. $\mathrm{O}(c_d^2(p^*))$ as well. A detailed calculation of this geometric series is given in [11]. Applying Lemma 5.3 now concludes the proof. □

We now give a matching lower bound to the upper bound of the last theorem.

Theorem 5.6. *Let the cost of a best route p^* for a given source destination pair with respect to a cost function $c(\cdot)$ on a unit disk graph G be $c(p^*)$. Then any (deterministic or randomized) geometric ad-hoc routing algorithm has expected cost $\Omega(c^2(p^*))$.*

Proof. We can directly apply the lower bound of [11] (Theorem 5.1). Because all edges (but one) of the graphs of [11] have length 1, the lower bound holds for all cost functions $c(\cdot)$. □

Combining Theorems 5.5 and 5.6 yields this section's main theorem, which states that AFR is asymptotically optimal on bounded degree unit disk graphs.

Theorem 5.7. *Let G be a bounded degree unit disk graph. Applying AFR on the Gabriel Graph edges of G to route from a given source to a given destination is asymptotically optimal among all possible geometric ad-hoc routing algorithms and for all cost functions.*

Proof. Combination of Theorem 5.5 and of Theorem 5.6. $\square$

6 General Unit Disk Graphs

In this section we consider the problem of geometric ad-hoc routing on general unit disk graphs (i.e. of unbounded degree). We describe a distributed algorithm which constructs a connected dominating set on an arbitrary unit disk graph G inducing a bounded degree unit disk graph. This subgraph of G is afterwards used as a routing backbone. We show that the resulting algorithm is asymptotically optimal for linearly bounded cost functions and that there is no geometric ad-hoc routing algorithm whose cost is bounded by the cost of an optimal path for super-linear cost functions.

6.1 Linearly Bounded Cost Functions

The connected dominating set algorithm, which we briefly review here for completeness, has been presented in [17].

The algorithm consists of two phases: (1) finding dominators and (2) finding connectors. The first phase results in a dominating set of the underlying graph, i.e. a set of dominators, such that each node in the network is either a dominator or the neighbor of at least one dominator. According to the dominator finding algorithm, no two dominators are neighbors. In order to produce a connected dominating set, the second phase is dedicated to finding unique nodes connecting pairs of dominators. The dominators together with the connectors finally form the routing backbone.

The first phase is realized by having each node u try to broadcast a message `IAmDominator(u)` to its direct neighbors telling them that it is a dominator. u may however only send this message if it has not received such a message beforehand by any of its neighbors. Each node stores all its dominators into a set `Dominators`. This algorithm leads to a dominating set whose members can be connected by bridges consisting of at most two intermediate nodes.

For the second phase, we use three types of messages:

- `IAmDominatee(u,v)` tells u's neighbors that it is dominated by v, i.e. that its neighbor v is a dominator.
- `2HopPath(u,v,w)` informs u's neighbors that it has a path to w via their connector v.
- `3HopPath(u,v,w,x)` tells u's neighbors that it has a path to w via their connectors v and w.

In addition to `Dominators` each node u keeps two sets:

- The path set `2HopConnectors` containing a path (w, v) iff w has been chosen as the connector to v.
- The path set `3HopConnectors` containing a path (w, v, x) iff w and v have been chosen as the connectors to x.

The second phase of the backbone construction algorithm is initiated by each dominatee u broadcasting `IAmDominatee(u,v)` for every v in its set `Dominators`. Afterwards, all network nodes react to received messages in the following way:

- A node u receives `IAmDominatee(v,w)` for the first time: If $u \neq w$, $v \notin$ `Dominators`, and no path of the form $(\cdot, w)$ is in `2HopConnectors`, u adds (v, w) to `2HopConnectors`. It broadcasts `2HopPath(u,v,w)` and will ignore any later messages `IAmDominatee(·,w)`. Note that a dominator broadcasts `2HopPath(u,v,w)` in order to signal v that it has been chosen as the connector between u and w, whereas a dominatee does so to enable the construction of 3-hop paths.
- A dominatee v receives `2HopPath(u,v,w)`: If $u \in$ `Dominators`, v becomes a connector.

- A dominator u receives `2HopPath(v,w,x)`: If neither `2HopConnectors` nor `3HopConnectors` contains a path to x, u adds (v, w, x) to `3HopConnectors` and broadcasts `3HopPath(u,v,w,x)`.

- A dominatee u receives `3HopPath(u,v,w,x)`: u becomes a connector and asks w to become a connector, too.

Note that the above algorithm may find a path connecting a dominator u with v differing from the path connecting v with u. This effect can be eliminated by introducing a total order on nodes (e.g. by node identifiers) and by allowing only a connection constructed from u to v if u is "smaller than" v.

The dominators together with the connectors form a backbone which can be employed for routing. In [17] the algorithm defines edges only along the found paths connecting dominators. The graph consisting of the backbone nodes and the found connecting edges is called CDS. What we are interested in, however, is the unit disk graph induced by all backbone nodes. We call this graph the *Backbone Graph.* Note that the edge set of the Backbone Graph is a superset of the edge set of CDS.

The following lemma describes the central property of the Backbone Graph for our purposes.

Lemma 6.1. *The Backbone Graph G_{BG} is a bounded degree unit disk graph.*

Proof. Since G_{BG} is a unit disk graph by definition, we only have to prove that it has bounded degree. By an area argument Lemma 1 of [17] shows that the number of dominators within a disk with radius k is upper-bounded by a constant $\ell_k \leq \frac{\pi(k+0.5)^2}{\pi(0.5)^2} = 4(k+0.5)^2$.

We now prove that any node u in G_{BG} has bounded degree. Since G_{BG} is a unit disk graph, u has an edge to every other node in CDS which lies within D_u, the unit disk centered at u. First, at most 5 dominators lie within D_u. Second, every connector has an edge to a dominator. Therefore, every node v within D_u has a dominating neighbor within the disk centered at u with radius 2. Since this disk contains no more than ℓ_2 dominators, and since every dominator has degree at most $2\ell_3$ in CDS (Lemma 4, [17]), the total number of nodes on CDS within D_u is upper-bounded by $2\ell_2\ell_3 + 5$. (Note that the factor 2 can be eliminated by halving a dominator's maximum degree via the introduction of an order over the nodes as suggested above.) □

Since the backbone contains a dominating set of the underlying graph, every regular node (a node not in the backbone) can be associated to one of its dominators. Since this can be regarded as a clustering of all regular nodes around their dominators, we call this graph the *Clustered Backbone Graph.* In order to route a message from a regular node s to a regular node t, the message will first be sent to s's associated dominator and then routed along the Backbone Graph to t's associated dominator before finally being forwarded to t itself. Note that while the Backbone Graph is bounded in degree, this is not the case for the Clustered Backbone Graph, since a dominator can have arbitrarily many dominatees. Also note that consequently in the Backbone Graph construction algorithm each node transmits at most a constant number of messages.

The following lemma shows that a route over the backbone is competitive with the optimal route for the link metric.

Lemma 6.2. *The Clustered Backbone Graph is a spanner with respect to the link metric, i.e. a best path between two nodes on the Clustered Backbone Graph is longer than a path between the same nodes in the underlying graph by a constant factor only.*

Proof. Follows directly from Lemma 5 in [17]. □

The next lemma shows that the Clustered Backbone Graph is a spanner with respect to all linearly bounded cost functions (see Section 4).

Lemma 6.3. *The Clustered Backbone Graph G_{CBG} is a spanner with respect to any linearly bounded cost metric $c(\cdot)$, i.e. the cost of an optimal path on G_{CBG} is only by a constant factor larger than the cost of an optimal path on the underlying unit disk graph G.*

Proof. Let $c_\ell(\cdot)$ be the link distance metric. By Lemma 6.2, we have a path p'_ℓ on G_{CBG} such that $c_\ell(p'_\ell) \in \Theta(c_\ell(p^*_\ell))$ where p^*_ℓ is an optimal link distance path on G. Let p^* denote an optimal path with respect to the cost $c(\cdot)$ on G. We then have to show that $c(p'_\ell) \in \mathrm{O}(c(p^*))$. The Euclidean length of p^* is $c_d(p^*)$ where $c_d(\cdot)$ denotes the cost function of the Euclidean distance metric. We partition p^* into maximal subpaths of length at most 1. Because two consecutive such subpaths have a total length greater than 1, we get at most $2 \cdot \lceil c_d(p^*) \rceil$ subpaths. We define the path p' by replacing each subpath with a direct edge. Note that all edges of p' have length at most 1. The link distance cost $c_\ell(p')$ of p' is upper-bounded by $c_\ell(p') \leq 2c_d(p^*) + 1$. By the optimality of p^*_ℓ, we also have $c_\ell(p') \geq c_\ell(p^*_\ell) \in \Theta(c_\ell(p'_\ell))$. And because with respect to the metric $c(\cdot)$, each edge of p'_ℓ has cost at most $c(1)$, we have $c(p'_\ell) \leq c(1)c_\ell(p'_\ell)$. Together, we get

$$c(p'_\ell) \in \mathrm{O}(c_d(p^*))\,. \tag{6}$$

Note that $c(1)$ is a constant because $c(x)$ has to be defined for all $x \in \,]0,1]$. Since $c(\cdot)$ has to be a linearly bounded cost function, the value

$$\alpha := \min_{x \in]0,1]} \frac{c(x)}{c_d(x)} = \min_{x \in]0,1]} \frac{c(x)}{x} > 0 \tag{7}$$

is a constant greater 0. Therefore $c(p^*) \geq \alpha c_d(p^*)$ and combined with Equation (6), we have

$$c(p'_\ell) \in \mathrm{O}(c(p^*))\,.$$

$\square$

We are now ready to apply AFR on general unit disk graphs. In a precomputation phase the Clustered Backbone Graph and its intersection with the Gabriel Graph are constructed. Then the routing from source s to destination t works as follows.

- If s and t are neighbors in G (the unit disk graph), the message is directly sent from s to t; otherwise, s sends the message to one of its dominators if s is not a dominator itself.
- Then we use AFR to route the message along the Gabriel Graph edges of the Clustered Backbone Graph. As soon as we arrive at a node whose Euclidean distance to t is at most one, the message is directly sent to t. Note that there has to be such a node on the boundary of one of the faces we visit.

Theorem 6.4. *Let the cost of the best path between a given source destination path with respect to a given linearly bounded cost metric be c. The cost of AFR as described above with respect to the same metric then is $\mathrm{O}(c^2)$.*

Proof. The case where s and t are direct neighbors follows from the fact that the cost function has to be linearly bounded and particularly from Equation (7). For the other cases we use that the Clustered Backbone Graph is a spanner for linearly bounded cost functions and that AFR has the given worst case cost on all bounded degree unit disk graphs (Theorem 5.5). $\square$

Theorem 6.5. *Applying AFR on the Gabriel Graph part of the Clustered Backbone Graph as described above is asymptotically optimal among all possible geometric ad-hoc routing algorithms for linearly bounded cost metrics.*

Proof. Because the lower bound of Theorem 5.6 holds for general unit disk graphs, Theorem 6.5 directly follows from Theorems 5.6 and 6.4. $\square$

6.2 Super-Linear Cost Functions

For the remainder of this section we consider geometric ad-hoc routing on general unit disk graphs for super-linear cost functions. Unlike for linearly bounded cost functions, the cost of a geometric ad-hoc routing algorithm cannot be bounded by the cost of an optimal path in this case.

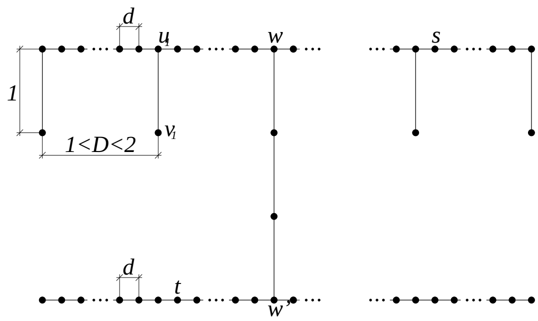

Figure 5: Lower bound graph for super-linear cost functions

Theorem 6.6. *Let the best route with respect to a super-linear cost function $c(\cdot)$ for a given source destination pair be p^*. Then, there is no (deterministic or randomized) geometric ad-hoc routing algorithm whose cost is bounded by a function of $c(p^*)$.*

Proof. We construct a family of unit disk graphs in the following way (see Figure 5). We choose a positive integer n and place $n+1$ nodes on a straight (say horizontal) line such that two neighboring nodes have distance $0 < d < 1$. Starting with the first node, we mark every $\lfloor 2/d \rfloor^{\text{th}}$ node. For every marked node u_i we then place a node v_i such that $\overline{u_i v_i}$ has length 1 and such that all the new nodes lie on a line which is parallel to the line where we put the first $n+1$ nodes. This yields k vertical edges of length one. The distance between two such edges is $D = \lfloor 2/d \rfloor d$. Note that $1 < D \leq 2$ because we have chosen d to be smaller than 1. The number of marked nodes (i.e. the number of such edges) k is then bounded by

$$k = \lfloor \frac{dn}{D} \rfloor \geq \lfloor \frac{dn}{2} \rfloor > \frac{dn}{2} - 1. \tag{8}$$

Now we choose an arbitrary marked node (we call it w) and the corresponding v_i. At v_i we add two other vertical edges and arrive at node w' which has distance 3 from the line with the original $n+1$ nodes. Symmetrically to the the original $n+1$ nodes, we now place another $n+1$ nodes (including w') on a horizontal line with distance 3. Figure 5 illustrates this construction.We choose an arbitrary node of the top $n+1$ nodes for the source s. The destination t is chosen arbitrarily from the bottom $n+1$ nodes. The optimal route p^* from s to t then first goes from s to w, then from w to w' and finally from w' to t. The cost of p^* can be bounded by

$$c(p^*) \leq 2nc(d) + 3c(1).$$

We want this cost to be constant and therefore choose $c(d) = 1/n$, yielding $d = c^{-1}(1/n)$. Note that since $c(\cdot)$ has to be nondecreasing, $c^{-1}(\cdot)$ is well-defined as long as there are no intervals where $c(\cdot)$ is constant. For those intervals we define $c^{-1}(\cdot)$ to take any of the possible values. For the cost of the optimal path $c(p^*)$ we now get a constant value ($c(1)$ is a constant!), i.e. $c(p^*) \in \Theta(1)$. In order to get the cost of a geometric ad-hoc routing algorithm $\mathcal{A}$, we observe that $\mathcal{A}$ has no information about the location of w and therefore has to test all possible nodes by using the k edges of length 1. For a deterministic $\mathcal{A}$ we can always place w such that it is the last marked node which is tried. For a randomized $\mathcal{A}$ we can place w such that the expected number of needed trials is at least $k/2$. For the cost $c(\mathcal{A})$ of any geometric ad-hoc routing algorithm we therefore get $c(\mathcal{A}) \in \Omega(k)c(1) = \Omega(k)$.

Plugging $d = c^{-1}(1/n)$ into Equation (8), we get

$$k \geq \frac{1}{2} n c^{-1}(1/n) - 1,$$

and for n approaching infinity we then get

$$\begin{aligned} \lim_{n\to\infty} k &\geq \lim_{n\to\infty} \frac{1}{2} n c^{-1}(1/n) - 1 \\ &= \frac{1}{2} \lim_{y\to 0} \frac{c^{-1}(y)}{y} - 1 \\ &= \frac{1}{2} \lim_{x\to 0} \frac{x}{c(x)} - 1 \;=\; \infty, \end{aligned}$$

where we substituted $y := 1/n$ in the first step and $x := c^{-1}(y)$ in the second step. The last limit is ∞, because by the definition of $c(\cdot)$, which is a super-linear cost function. We have $\lim_{x\to 0} c(x)/x = 0$. Therefore, the cost of any algorithm $\mathcal{A}$ is unbounded with respect to the best path p^*, which has constant cost. $\square$

7 Conclusion

In this paper we introduced a general definition of cost metrics for mobile ad-hoc routing. We showed that the cost metrics are naturally partitioned into *linearly bounded* and *super-linear* cost functions according to their behavior for an edge approaching zero length. In particular, we constructed graphs for which there is no geometric routing algorithm that finds a route whose cost is bounded by the cost of the optimal route if the cost metric is in the second class. On the other hand, a linearly bounded cost metric allows to find an algorithm with guaranteed message delivery whose cost is quadratic in the cost of the optimal route, which at the same time is shown to be a tight upper bound. We showed that on a *bounded degree unit disk graph* this bound holds also for super-linear cost metrics. Finally, we proposed an algorithm that reaches the optimum by construction of a network backbone and by employment of *Adaptive Face Routing*.

We implicitly analyze a one-shot scenario, where routing takes place once the backbone is formed for a given network topology. An interesting open question is how to maintain the backbone in a dynamic network efficiently.

References

[1] P. Bose and P. Morin. Online routing in triangulations. In *10th International Symposium on Algorithms and Computation (ISAAC)*, volume 1741 of Springer LNCS, pages 113–122, 1999.

[2] P. Bose, P. Morin, A. Brodnik, S. Carlsson, E. Demaine, R. Fleischer, J. Munro, and A. Lopez-Ortiz. Online routing in convex subdivisions. In *International Symposium on Algorithms and Computation (ISAAC)*, pages 47–59, 2000.

[3] C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks. In *The IEEE Singapore International Conference on Networks*, pages 197–211, 1997.

[4] G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, USC/ISI, March 1987.

[5] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. In J. Díaz and M. Serna, editors, *Algorithms—ESA '96, Fourth Annual European Symposium*, volume 1136 of *Lecture Notes in Computer Science*, pages 179–193, Barcelona, Spain, 25–27 Sept. 1996. Springer.

[6] T. Hou and V. Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications*, 34(1):38–44, 1986.

[7] T. Imielinski and J. Navas. Gps-based addressing and routing. Technical Report RFC 2009, Computer Science, Rutgers University, November 1996.

[8] Y. Ko and N. Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms, 1998.

[9] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. 11th Canadian Conference on Computational Geometry*, pages 51–54, Vancouver, August 1999.

[10] P. Krishna, M. Chatterjee, N. H. Vaidya, and D. K. Pradhan. A cluster-based approach for routing in ad-hoc networks. In USENIX Association, editor, *Proceedings of the second USENIX Symposium on Mobile and Location-Independent Computing: April 10–11, 1995, Ann Arbor, Michigan, USA*, pages 1–10, Berkeley, CA, USA, Apr. 1995. USENIX.

[11] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. Technical Report 369, ETH Zurich, Department of Computer Science, 2002. Submitted. http://www.distcomp.ethz.ch/publications/tr369.ps.

[12] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad-hoc routing. In *MobiCom '00*, pages 120–130, August 2000.

[13] J. Navas and T. Imielinski. Geocast - geographic addressing and routing. In *Mobile Computing and Networking*, pages 66–76, 1997.

[14] A. Savvides, C.-C. Han, and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MOBICOM '01*, pages 166–179, July 2001.

[15] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[16] J. Urrutia. Routing with guaranteed delivery in geometric and wireless networks. In I. Stojmenovic, editor, *Handbook of Wireless Networks and Mobile Computing*, chapter 18, pages 393–406. John Wiley & Sons, 2002.

[17] Y. Wang and X.-Y. Li. Geometric spanners for wireless ad hoc networks. In *22^{nd} International Conference on Distributed Computing Systems*, 2002. http://www.iit.edu/~wangyu1/research/spanner-ICDCS.ps.

[18] J. Wu. Dominating-set-based routing in ad hoc wireless networks. In I. Stojmenovic, editor, *Handbook of Wireless Networks and Mobile Computing*, chapter 20, pages 425–450. John Wiley & Sons, 2002.