Geometric Ad-Hoc Routing: Of Theory and Practice

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

Abstract

All too often a seemingly insurmountable divide between theory and practice can be witnessed. In this paper we try to contribute to narrowing this gap in the field of ad-hoc routing. In particular we consider two aspects: We propose a new geometric routing algorithm which is outstandingly efficient on *practical* average-case networks, however is also in *theory* asymptotically worst-case optimal. On the other hand we are able to drop the formerly necessary assumption that the distance between network nodes may not fall below a constant value, an assumption that cannot be maintained for *practical* networks. Abandoning this assumption we identify from a *theoretical* point of view two fundamentamentally different classes of cost metrics for routing in ad-hoc networks.

Keywords: Mobile computing, ad-hoc networks, routing, cost metrics.

1 Introduction

An ad-hoc network consists of mobile nodes equipped with radio devices. If the source and the destination of a message are not within mutual transmission range, the message can be relayed by intermediate nodes, a process known as *ad-hoc routing*. In this paper we study *geometric* routing, which assumes a) that each network node is informed about its own and about its neighbors' positions and b) that the source of a message knows the position of the destination. The employment of position information becomes more and more realistic with increasing availability of inexpensive positioning systems. The same goal could also be achieved by local information exchange with fixed beacon nodes. Similarly the location of the destination could be learned via an overlay (e.g. peer-to-peer [21, 27]) information system. But also a scenario is conceivable, where a message needs to be sent to *any* node in a given area (also called "geocasting" [16, 22]). Since none of the intermediate nodes is required to maintain routing lists, geometric routing can be considered a lean version of source routing [14].

Our geometric routing algorithm GOAFR$^+$ (pronounced as "gopher-plus") combines—similarly to earlier proposals [4, 6, 15, 20]—two concepts called greedy routing and face routing. In greedy routing mode the algorithm forwards the routed message at each network node to the neighbor closest to the destination. Already in simple configurations, the message can however reach a "dead end", a node without any "better" neighbor. Such cases are overcome by the employment of face routing, which explores the boundaries of faces of the planarized network graph. GOAFR$^+$ uses an "early fallback" technique to return to greedy routing as soon as possible. Our simulations show that—additionally restricting its search to an adaptively resized area—the algorithm is even more efficient than similar algorithms analyzed earlier on average (random) graphs. On the other hand our theoretical analysis proves that GOAFR$^+$ is asymptotically optimal in the worst case.

Theoretical analysis of routing algorithms often has to make irritating or far-fetched assumptions, which would hardly ever hold in practice. In this paper we are able to drop one such assumption, the $\Omega(1)$-model introduced in [19], which assumes that the distance between network nodes cannot fall beneath a constant minimum bound. Graphs with this restriction have also been called *civilized* [7] or λ-*precision* [13] graphs in the literature. We introduce a general notion of a cost metric, defined as a nondecreasing function of the length of the edge over which a message is sent. We show that the behavior of cost functions for edge length approaching zero proves crucial for the cost of routing. We observe that in theory cost metrics fall into two classes: *Linearly bounded* cost functions are bounded from below by a linear function; for *super-linear* functions such a bounding linear function does not exist. With cost metrics from the former class, a clustering technique allows the construction of a routing backbone, which extends GOAFR$^+$'s asymptotic optimality to networks with nodes of arbitrarily small distance. With cost functions from the latter class on the other hand an example graph can be constructed for which there exists no geometric routing algorithm whose execution cost is competitive with the cost of the optimal path.

After giving an overview of related work in the following section, we state the model used in this paper in Section 3. In Section 4 we introduce our routing algorithm GOAFR$^+$, prove its asymptotic optimality, and present simulation results. Section 5 introduces a definition of general cost metrics for routing, identifies two classes of metrics, *linearly bounded* and *super-linear*, and describes the consequences of this classification on the cost of routing. Section 6 finally summarizes the paper.

2 Related Work

The early proposals of geometric routing—suggested over a decade ago—were of purely greedy nature: At each intermediate network node the message to be routed is forwarded to the neighbor closest to the destination [8, 12, 23]. This can however fail if the message reaches a local minimum with respect to the distance to the destination, that is a node without any "better" neighbors. Also a "least deviation angle" approach (*Compass Routing* in [17]) cannot guarantee message delivery in all cases.

The first geometric routing algorithm that does guarantee delivery was *Face Routing* introduced in [17] (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 [4, 6]; 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 [2]. [3] proposed an algorithm competitive with the shortest path between source and destination on Delaunay triangulations; this is however not applicable to ad-hoc networks, since Delaunay triangulations may contain arbitrarily long edges, whereas transmission ranges are limited. Accordingly [10] proposed local approximation of the Delaunay Graph, however without improving performance bounds for routing. A more detailed overview of geometric routing can be found in [24].

In [19] we proposed *Adaptive Face Routing* AFR. The execution cost of this algorithm—basically enhancing Face Routing by the employment of an ellipse restricting the searchable area—is bounded by the cost of the optimal route. In particular, the cost of AFR is not greater than the squared cost of the optimal route. We also showed that this is the worst-case optimal result any geometric routing algorithm can achieve.

Face Routing and also AFR are not applicable for practical purposes due to their strict employment of face traversal. There have been proposals for practical purposes to combine greedy routing with face routing [4, 6, 15], however without competitive worst-case guarantees. In [20] we suggested, to the best of our knowledge, the first algorithm to combine greedy and face routing in a worst-case optimal way; in order to remain asymptotically optimal, this algorithm could however not include falling back as soon as possible from face to greedy routing, an obvious improvement for the average case performance.

In this paper we use a clustering technique in order to drop the $\Omega(1)$-model assumption from [19]. Clustering for the means of ad-hoc routing has been proposed by various researchers [5, 18]. A closely related approach is the construction of *connected dominating sets* as routing backbones [11, 26].

3 Model and Preliminaries

In this paper we assume that network nodes are placed in the Euclidean plane $\mathbb{R}^2$. In order to represent ad-hoc networks we adopt the widely used model, where every node has the same transmission range, without loss of generality normalized to 1. The resulting graph, having an edge between two nodes u and v iff the Euclidean distance $|\overline{uv}| \leq 1$, is a *unit disk graph.*

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

Definition 3.1. (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 the above metrics.

For convenience we also define the cost of paths, a sequence of contiguous edges, and algorithms. The cost $c(p)$ of a path p is defined as the sum of the costs of its edges. 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.

For our routing algorithm the network graph is required to be *planar*, that is without intersecting edges. For this purpose we employ the *Gabriel Graph*. A Gabriel Graph (on a given node set in the Euclidean plane) is defined to contain an edge between two nodes u and v iff the circle having $\overline{uv}$ as a diameter does not contain a witness node w. This graph features two important properties: a) It can be computed locally (each node merely inspecting its neighbors' positions) and b) its construction on G preserves an energy-minimal path between any pair of network nodes, which—by equivalence of cost metrics (Section 5.1)—entails that the construction of the Gabriel Graph on G's nodes also preserves G's distance properties up to constants.

In our analysis we use the concept of a unit disk graph whose nodes do not have more than a constant number of neighbors. A unit disk graph G is a *bounded degree unit disk graph with parameter k* if none of its nodes has degree greater than k.

We consider *geometric routing algorithms* [19]. The aim of the algorithm is to forward a message from a given source s to a given destination t over the edges of the network graph while complying with the following rules:

- Each node knows its own and its neighbors' positions.

- The source s is informed about the destination t's position.

- A node is allowed to store only local information or temporarily present packets in transit.

- A packet may contain control information about at most O(1) nodes.

According to these rules geometric routing algorithms are inherently of local nature.

Finally we assume routing to take place much faster than node movement: A routing algorithm executes on temporarily stationary nodes.

4 GOAFR$^+$

In this section we introduce the GOAFR$^+$ (pronounced as "gopher-plus") algorithm. We prove that the algorithm is asymptotically optimal if the network graph is a bounded degree unit disk graph. The construction of a bounded degree unit disk graph from a general unit disk graph will be discussed in Section 5.2.1. Our simulation results show that GOAFR$^+$ is also efficient on average case graphs.

4.1 The GOAFR$^+$ Algorithm

The GOAFR$^+$ algorithm is a combination of *greedy routing* and *face routing*. Whenever possible the algorithm tries to route greedily, that is by forwarding the message at each intermediate node to the neighbor located closest to the destination t. Doing so, however, the algorithm can reach a *local minimum* with respect to the distance from t, that is a node u_m none of whose neighbors is located closer to t than u_m itself.

In order to overcome such a local minimum, GOAFR$^+$ applies a face routing technique, borrowing from the *Face Routing* algorithm originally introduced in [17]. Face Routing proceeds towards the destination by exploring the boundaries of the faces of a planarized network graph, employing the local *right hand rule* (in analogy to following the right hand wall in a maze). Additionally the algorithm restricts itself to a searchable area occasionally being resized during algorithm execution. With this approach the algorithm becomes asymptotically optimal with respect to its execution cost compared with the cost of the optimal path. A similar concept was introduced in [19].

Having escaped the local minimum, the algorithm continues in greedy mode. Since greedy forwarding is—above all in dense networks—more efficient than face routing in the average case, the algorithm should for practical purposes fall back to greedy mode as soon as possible. In [20] we studied a family of similar algorithms combining greedy and face routing. We observed that algorithm variants with heuristics employed for early fallback to greedy mode (such as the "First Closer" heuristic having the algorithm resume greedy routing as soon as meeting a node closer to the destination than where the current face routing phase started) lose their asymptotic optimality with respect to the shortest path. It appeared that, once in face routing mode, an algorithm is required to explore the *complete* boundary of the current face in order to be asymptotically optimal.

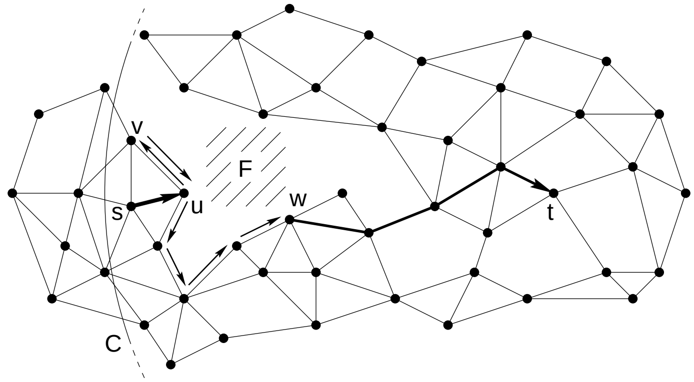

Figure 1: The GOAFR$^+$ algorithm starts from s in greedy mode. At node u it reaches a local minimum, a node without any neighbors closer to t. GOAFR$^+$ switches to face routing mode and begins to explore the boundary of face F (in clockwise direction). At node v the algorithm hits the bounding circle C and turns back to continue the exploration of F's boundary in the opposite direction. After each step the counters p and q are updated. At node w the fallback condition $p > \sigma\, q$ holds ($p = 2, q = 4$ with the assumption $1/4 \leq \sigma < 1/2$); GOAFR$^+$ falls back to greedy mode and continues to finally reach t. (Gradual reduction of C's size during GOAFR$^+$'s execution is not shown.)

Contrarily to this conjecture, the GOAFR$^+$ algorithm does not necessarily explore the complete face boundary in face routing mode and yet *does* conserve asymptotic optimality. For this purpose the algorithm employs two counters p and q to keep track of how many of the nodes visited during the current face routing phase are located closer (p) and how many are not closer (q) to the destination than the starting point of the current face routing phase; as soon as a certain fallback condition holds, GOAFR$^+$ directly falls back to greedy mode. Besides being asymptotically optimal, however, simulations show that in the average case GOAFR$^+$ even outperforms the best (not asymptotically optimal!) algorithms considered in [20].

In particular GOAFR$^+$ consists of the following steps:

GOAFR$^+$ The algorithm parameters ρ_0, ρ, and σ are chosen prior to algorithm start and remain constant throughout the execution. For the algorithm to work correctly, they have to comply with the conditions $1 \leq \rho_0 < \rho$ and $0 < \sigma$.[1]

0. Begin at s. Initialize C to be the circle centered at t with radius $r_C := \rho_0\, |\overline{st}|$.

1. **(Greedy Routing Mode)** Repeat taking greedy steps until either reaching t or a local minimum. In the former case the algorithm terminates, in the latter case continue with step 2. Whenever possible, reduce C's radius ($r_C := r_C/\rho$) as long as the currently visited node stays within C.

2. **(Face Routing Mode)** Let u_i be the currently visited local minimum. Start exploring the boundary of F_i, the face containing the connecting line $\overline{u_i t}$ in the immediate environment of u_i. When completing F_i's exploration and returning to u_i, advance to the node visited so far closest to t and continue with step 1. If no visited node is closer to t than u_i, report graph disconnection to s (using GOAFR$^+$). During the exploration of F_i's boundary use two counters p and q to keep track of the number of nodes visited on F_i's boundary: p counts the nodes closer to t than u_i and q the nodes not located closer to t than u_i. Take a special action if one of the following conditions holds:

   2a. Hitting C for the first time, turn back and continue exploring F_i's boundary in the opposite direction.

   2b. C is hit for the second time: If none of the visited nodes is closer to t than u_i, enlarge C ($r_C := \rho\, r_C$) and continue with step 2 as if started from u_i. Otherwise advance to the node visited so far closest to t and continue with step 1.

   2c. If $p > \sigma\, q$, that is, we have visited (up to a constant factor σ) more nodes on F_i's boundary closer to t than nodes not closer to t, advance to the node seen so far closest to t (if this is not the currently visited node) and continue with step 1.

[1] In our simulations $\rho_0 = 1.4$, $\rho = \sqrt{2}$, and $\sigma = \frac{1}{100}$ proved to be good choices for practical purposes.

4.2 GOAFR$^+$ is Asymptotically Optimal

In the following we prove that GOAFR$^+$ is asymptotically optimal on *bounded degree unit disk graphs.* In Section 5.1 we will prove that on bounded degree unit disk graphs all cost metrics (defined according to Definition 3.1) are equivalent up to constants. In Section 5.2 we will show that such a graph can be constructed from a general unit disk graph (that is of unbounded degree). By these means GOAFR$^+$ can be extended to perform asymptotically optimally on general unit disk graphs for a certain class of cost metrics.

The GOAFR$^+$ algorithm runs on a *planar* graph. As mentioned in Section 3 we employ the Gabriel Graph for this purpose. In our analysis we therefore assume GOAFR$^+$ to run on G_{GG}, the intersection of the bounded degree unit disk graph G and the corresponding Gabriel Graph.

We begin the analysis of GOAFR$^+$ by stating a fact on the number of nodes in a given two-dimensional region:

Lemma 4.1. *Let $R \subset \mathbb{R}^2$ be a 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*

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

Proof. Cf. Appendix A. □

GOAFR$^+$ uses a circle C centered at t to restrict itself to a searchable area. During the algorithm execution the radius r_C is adapted in predefined steps according to the current distance from t. In particular, the values potentially assumed by r_C form a geometric sequence $r_{C_i} = r_{max}\,(\frac{1}{\rho})^i, i = 0...k$, where r_{max} depends on the length and the shape of the optimal path from s to t (cf. proof of Theorem 4.5) and ρ is one of GOAFR$^+$'s predefined constant algorithm parameters. Since r_C can both increase and decrease during algorithm execution, the steps taken in a circle C_i with radius r_{C_i} need not occur consecutively. In the following we consider the steps taken by the algorithm in a fixed circle C_i.

Lemma 4.2. *If s and t are connected within the circle C_i, GOAFR$^+$ reaches t. If s and t are not connected, GOAFR$^+$ reports so.*

Proof. Cf. Appendix B. □

For the following lemma we define a *round* according to the algorithm to be either a) a greedy step, b) a face routing phase terminated by early fallback, or c) a face routing phase terminated after exploration of the complete boundary of the current face and advancing to the node closest to t. In Appendix B we show that after each round the algorithm is strictly closer to t than before that round.

Lemma 4.3. *Let $c'_F(GOAFR^+)$ be the cost of all face routing steps taken when exploring the boundary of face F within the circle C_i. $c'_F(GOAFR^+)$ is less than $\gamma\, c_F$ for a constant γ and c_F being the total cost of traversing F's boundary once.*

Proof. We first show that the lemma holds for the link distance metric, $c(e) \equiv 1$ for any edge e: The total number of edges traversed by GOAFR$^+$ when exploring F is less than γc_{ℓ_F}, where c_{ℓ_F} is the number of edges traversed when traveling around F once.

We introduce directed edges or *arcs* and say that the algorithm traverses the arc from u to the *target* v whenever the algorithm traverses an edge from node u to node v. We denote the target node of an arc e by e_t. We assume that the boundary of face F is involved in k face routing rounds, and that for $1 \leq j \leq k$, s_j is the node where round j is started. T_j is the set of all arcs visited in round j. Furthermore we define $P_j := \{e \in T_j : |\overline{e_t t}| < |\overline{s_j t}|\}$ and $Q_j := \{e \in T_j : |\overline{e_t t}| \geq |\overline{s_j t}|\}$ (cf. counters in GOAFR$^+$ algorithm). Finally O_j is the set of "old" edges already traversed (in either direction) in any of the earlier rounds and N_j is the set of edges newly traversed (again in either direction) in round j. Since after each round—a greedy step or the exploration of a face—the algorithm is strictly closer to t than before that round, all old edges are not adjacent to a target closer to t than s_j: $|O_j| \leq |Q_j|$. Since all arcs in P_j have not been taken before, we have $|P_j| \leq 5\,|N_j|$. (The constant 5 is introduced, since a) an *undirected* edge in N_j can be traversed for a second time after the algorithm has hit C_i, b) the same face F can lie on both sides of an edge, and c) the edge can be traversed once more during the algorithm's advancing to the node seen so far closest to t after the fallback condition holds.) According to the fallback condition in the algorithm, we have $|P_j| > \sigma\,|Q_j|$. In summary we can conclude $|O_j| \leq |Q_j| < |P_j|/\sigma \leq 5\,|N_j|/\sigma$. With $|T_j| \leq 5\,(|O_j| + |N_j|)$ (the constant 5 appears for the same reason as above) we obtain for the total cost of the algorithm on F: $\sum_{j=1}^{k} |T_j| \leq \sum_{j=1}^{k} 5\,(|O_j|+|N_j|) < 5\sum_{j=1}^{k}(5\,|N_j|/\sigma+|N_j|) = 5\,(1+5/\sigma)\sum_{j=1}^{k}|N_j| \leq 5(1+5/\sigma)c_{\ell_F}$, the last step following from $\sum_{j=1}^{k}|N_j| \leq c_{\ell_F}$. (A smaller constant could be obtained by a more intricate analysis.)

If the fallback criterion never holds during F's exploration (which is only possible in the final round for F), the algorithm traverses F's complete boundary and advances to the node closest to t, which incurs additional cost less than $2\,c_F$.

The lemma holds for the link distance metric. Since the algorithm is assumed to run on a bounded degree unit disk graph, the lemma also holds for any other cost metric (cf. Section 5.1). □

Lemma 4.4. *The total cost of the steps taken by GOAFR$^+$ within the circle C_i with radius r_{C_i} is in* $\mathrm{O}(r_{C_i}^2)$.

Proof. According to the previous lemma we have $c'_F(\mathrm{GOAFR}^+) \le \gamma\, c_F$ for all steps performed in face routing mode. Summing up over all faces in C_i we obtain $\sum_{F\in C_i} c'_F(\mathrm{GOAFR}^+) < \gamma \sum_{F\in C_i} c_F \le \gamma \cdot 2\sum_{e\in C_i} c(e)$, the last step following from the fact that each edge e is adjacent to at most two faces. To account for the greedy steps we add another $\sum_{e\in C_i} c(e)$, since any edge can be traversed at most once in greedy mode (each round—a greedy step or the exploration of a face—taking the algorithm strictly closer to t). Since we employ a planar graph, with the fact that (in a graph with more than three edges) each face is adjacent to at least three edges and using the Euler polyhedral formula we obtain that $|E_i| \in \mathrm{O}(|V_i|)$, where $|E_i|$ is the number of edges and $|V_i|$ the number of nodes in C_i. The lemma finally follows with $\sum_{e\in C_i} c(e) \in \mathrm{O}(|E_i|)$—resulting from the equivalence of the link distance metric with any other metric on bounded degree unit disk graphs (cf. Section 5.1)—and Lemma 4.1. □

As described above, GOAFR$^+$ employs a set of bounding circles whose radii form a geometric sequence. This together with the fact that the maximum radius is bounded by the Euclidean length of an optimal path from s to t, leads to the following theorem.

Theorem 4.5. *Let p^* be an optimal path from s to t. On a bounded degree unit disk graph GOAFR$^+$ reaches t with cost $\mathrm{O}(c^2(p^*))$, if s and t are connected, which is asymptotically optimal. If s and t are not connected, GOAFR$^+$ reports so to the source.*

Proof. Let $c_\ell(p^*)$ be the Euclidean length of a shortest path from s to t. If s and t are connected, the circle centered at t and with radius $c_\ell(p^*)$ completely contains p^*. Since GOAFR$^+$ only enlarges the bounding circle if it does not contain a path from s to t, and according to GOAFR$^+$'s radius update policy with the constant factor ρ, the maximum radius reached is smaller than $\rho\, c_\ell(p^*)$. In order to compute the total cost of the algorithm we add up the cost expended in each used circle. According to Lemma 4.4 and Lemma 4.1 it is sufficient to consider the areas of all employed circles. Let r_{max} be the radius of the largest used circle. For some $k \ge 0$ the areas of all used circles sum up to $\sum_{i=0}^{k} \pi(r_{max} \cdot \frac{1}{\rho^i})^2 = \frac{1-1/\rho^{2(k+1)}}{1-1/\rho^2}\,\pi r_{max}^2 < \frac{1-1/\rho^{2(k+1)}}{1-1/\rho^2}\,\pi(\rho\, c_\ell(p^*))^2 \in \mathrm{O}(c_\ell(p^*)^2)$. With the equivalence of cost metrics—including the Euclidean metric—on bounded degree unit disk graphs, this holds for any metric. Asymptotic optimality follows from the lower-bound example in [19, Figure 8].

If s and t are not connected, GOAFR$^+$ detects so (case c) in proof of Lemma 4.2) and reports back to the source using the same algorithm. □

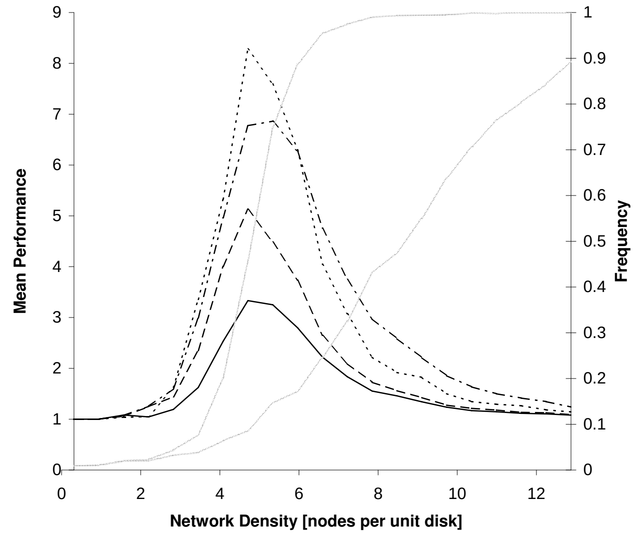

Figure 2: Performance of routing algorithms in critical network density range around 4.5 nodes per unit disk. Mean performance values for GOAFR$^+$ (solid line), GOAFR$_{\mathrm{FC}}$ (dashed), GOAFR (dash-dotted), and GPSR (dotted) plotted against the left y axis. The network connectivity and greedy success rate are plotted for reference (in gray against right y axis).

4.3 Average-Case Efficiency

The GOAFR$^+$algorithm includes greedy routing and an early fallback mechanism intended to reduce the algorithm cost on average case graphs. In order to assess the algorithm's average case performance we employed the custom simulation environment introduced in [20]. The simulations were carried out on graphs generated by randomly and uniformly placing nodes on a square field of side length 20 units and by randomly choosing a source-destination pair. In [20] we identified a critical network density range around 4.71 ($\approx 1.5\pi$) nodes per unit disk. Situated between low densities, where only in trivial cases s and t are connected at all, and high densities, where in most cases greedy routing will succeed in finding a good path, this density range forms a challenge to routing algorithms: Generally the length of the shortest path from the source to the destination is significantly longer than their (Euclidean) distance.

Figure 2 depicts the measured performance values of four routing algorithms around this critical network density. For each simulated network density the plotted performance value is the mean of the ratios between the algorithm cost and the cost of the shortest path (with respect to the link distance metric) measured on 2000 generated (network, source, destination) triples: Low performance values are rated good. The *network connectivity rate*—showing in how many of the generated networks s and t are connected—and the *greedy success rate*—representing how often the algorithm reaches t by employment of greedy routing alone—are depicted for reference and identification of the critical density range.

Figure 2 contains the performance values for the GPSR algorithm [15], for GOAFR and GOAFR_{FC} [20], as well as for GOAFR^+. The GPSR algorithm combines greedy and face routing, including early fallback, does however not employ the concept of a bounding searchable area. Making use of this concept, the GOAFR algorithm becomes asymptotically worst-case optimal, yet is not efficient in practice, since—once in face routing mode—always complete face boundaries are explored. In order to avoid this effect, an early fallback heuristic is applied by the GOAFR_{FC} algorithm. This algorithm showed best average-case performance in [20], is however not asymptotically worst-case optimal. GOAFR^+ in contrast shows clearly better performance values for the critical density range—exploiting successive reduction of the bounding area size—and at the same time is also asymptotically optimal in the worst case.

5 Cost Metric

In this section we discuss the properties of cost metrics defined according to Definition 3.1 in the context of geometric routing. We first show that all possible such cost metrics are equivalent up to constant factors on bounded degree unit disk graphs. In a second part we prove that when considering general unit disk graphs (without bounded degree) the cost functions are divided into two classes, *linearly bounded* and *super-linear*. We show that employing a backbone construction GOAFR^+'s optimality can be extended to general unit disk graphs for linearly bounded cost functions. With super-linear cost metrics on the other hand, a lower bound graph proves that there exists no geometric routing algorithm whose cost is bounded with respect to the shortest path.

5.1 Bounded Degree Unit Disk Graphs

For the proof of GOAFR^+'s asymptotic optimality on bounded degree unit disk graphs in Section 4.2 we employed the equivalence of all cost metrics on such graphs. This equivalence up to a constant factor is shown in the following lemma.

Lemma 5.1. *Let $c_1(\cdot)$ and $c_2(\cdot)$ be cost functions as defined in Definition 3.1 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 Sketch. The proof for this lemma exploits the fact that p is cycle-free and therefore, starting at a node u, we leave the disk with radius 1 around u after traversing at most $k+1$ edges. Thus any metric $c(p)$ can be bounded (up to constants) from above by the Euclidean metric $c_d(p)$. With a similar argument $c(p)$ can also be bounded from below by the Euclidean metric. Consequently we obtain $c(p) \in \Theta(c_d(p))$. The exact proof can be found in Appendix C. ☐

As an application of Lemma 5.1 we obtain the following lemma.

Lemma 5.2. *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. Cf. Appendix D. ☐

5.2 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). As shown in the following the behavior around 0 divides the cost functions defined according to Definition 3.1 into two natural classes. The cost functions lower-bounded by a linear function are called **linearly bounded cost functions**, the cost functions not bounded by a linear function are called **super-linear cost functions**.

$$\begin{aligned} \text{linearly bounded:} \quad & \exists\, m > 0 : \; c(d) \geq m \cdot d, \; \forall\, d \in]0,1], \\ \text{super-linear:} \quad & \nexists\, m > 0 : \; c(d) \geq m \cdot d, \; \forall\, d \in]0,1]. \end{aligned}$$

Of the standard cost measures the link distance and the Euclidean metric are linearly bounded, whereas the energy metric is super-linear. The lower bound example of Section 5.2.2 exploits the property that with super-linear cost functions it is possible to construct chains with nodes of distance approaching zero which allow to cover a finite Euclidean distance "for free" in the limit.

We now give an algorithm which is asymptotically optimal for linearly bounded cost functions. We subsequently show 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.

5.2.1 Linearly Bounded Cost Functions

First we describe our algorithm as it can be applied to an arbitrary unit disk graph G and for all linearly bounded costs. In a precomputation phase a routing backbone G_{BG} is calculated. G_{BG} is a subgraph of G such that a) G_{BG} is a bounded degree unit disk graph and b) the nodes of G_{BG} form a connected dominating set of G. Consequently, all nodes of G have at least one neighbor in G_{BG}. The distributed construction of a subgraph of G with properties a) and b) is described in a number of publications (e.g. [1, 9, 25]).

As 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* G_{CBG}. 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.

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

Lemma 5.3. *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 unit disk graph by a constant factor only.*

Proof. Follows from [25, Lemma 5]. □

This property of the Clustered Backbone Graph does not only hold for the link distance metric, but for all linearly bounded cost functions.

Lemma 5.4. *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 greater 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 5.3, 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 $\lceil 2\, c_d(p^*) \rceil$ 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

$$c(p'_\ell) \in \mathrm{O}(c_d(p^*))\,. \tag{1}$$

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, we have $c(x) \geq m \cdot c_d(x)$ for a constant $m > 0$. Therefore also $c(p^*) \geq m \cdot c_d(p^*)$, and combined with Equation (1) we obtain

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

□

Our routing algorithm GOAFR$^+$ works on planar graphs. 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. It is well known that the intersection between the Gabriel Graph and the unit disk graph (GG $\cap$ UDG) is connected iff the UDG is connected. It is also well known that GG $\cap$ UDG contains an energy optimal path (see Figure 7 in [19]). This leads to the next lemma.

Lemma 5.5. *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 5.2, we now see that the optimal energy path p^*_{E} is competitive for all cost functions $c(\cdot)$, i.e. $c(p^*_{\mathrm{E}}) \in \Theta(c(p^*))$. □

We are now ready to apply GOAFR$^+$ 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 GOAFR$^+$ 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 5.6. *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 GOAFR$^+$ as described above with respect to the same metric then is $\mathrm{O}(c^2)$. This is asymptotically optimal among all possible geometric ad-hoc routing algorithms for linearly bounded cost metrics.*

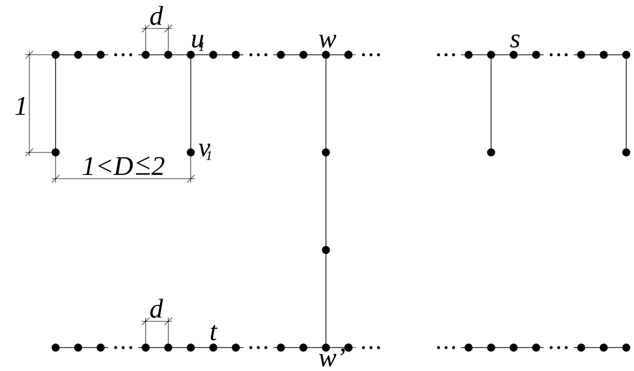

Figure 3: Lower bound graph for super-linear cost functions

Proof. The case where s and t are direct neighbors follows from the fact that the cost function has to be linearly bounded. For the other cases we use that the intersection of the Gabriel Graph and the Clustered Backbone Graph is a spanner for linearly bounded cost functions (Lemmas 5.4 and 5.5) and that GOAFR$^+$ has the given worst case cost on all bounded degree unit disk graphs (Theorem 4.5). Optimality follows from Theorem 4.5, since the $\Omega(c^2)$ lower bound graph is also a Clustered Backbone Graph. □

5.2.2 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.

Theorem 5.7. *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 3). 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

$$k = \left\lfloor \frac{dn}{D} \right\rfloor \geq \left\lfloor \frac{dn}{2} \right\rfloor > \frac{dn}{2} - 1. \qquad (2)$$

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 original $n+1$ nodes, we now place another row of $n+1$ nodes (including w') on a horizontal line with distance 3. Figure 3 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 (2), we get

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

and for n approaching infinity we then obtain

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

where we substituted $y := 1/n$ in the first step and $x := c^{-1}(y)$ in the second step. The last limit is ∞ by the definition of $c(\cdot)$, a super-linear cost function, which implies that $\lim_{x\to 0} c(x)/x = 0$ if this limit exists. (For convenience we assume that the limit exists. Otherwise the same result can be achieved by "tuning" the graph more closely to the cost function.) Therefore, the cost of any algorithm $\mathcal{A}$ is unbounded with respect to the best path p^*, which has constant cost. □

6 Conclusion

Trying to help bridging the chasm between theory and practice in the field of ad-hoc routing, we proposed in this paper the geometric routing algorithm GOAFR$^+$, which is more efficient than any previously studied algorithm on average case graphs, while being also in the worst case asymptotically optimal. We defined a general cost model for routing algorithms and observed that all possible cost functions fall into two classes, *linearly bounded* and *super-linear*. For linearly bounded cost functions GOAFR$^+$ could be extended such that the formerly necessary $\Omega(1)$-model restriction on node distances could be dropped. With super-linear cost functions an example graph was presented, for which there exists no geometric routing algorithm of cost competitive with the shortest path.

Of the most popular cost metrics—link distance (hop), Euclidean distance. and energy metric—the first two are linearly bounded, whereas the energy metric is super-linear. In practical wireless ad-hoc networks, however,—also in systems with adaptable transmission power—the energy required for the transmission of a message will never drop below a certain base energy even for minimum transmission distance. Consequently also for power-adaptive transmission the cost function will be linearly bounded. For all practical cost metrics it is therefore possible to drop the $\Omega(1)$-model assumption and still remain asymptotically optimal by employment of the backbone construction.

References

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks. In *Proc. ACM Int. Symposium on Mobile ad hoc networking & computing (MobiHoc)*, 2002.

[2] P. Bose, A. Brodnik, S. Carlsson, E.D. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, and J.I. Munro.

Online Routing in Convex Subdivisions. In *International Symposium on Algorithms and Computation (ISAAC)*, volume 1969 of *Lecture Notes in Computer Science*, pages 47–59. Springer, 2000.

[3] P. Bose and P. Morin. Online Routing in Triangulations. In *Proc. 10^{th} Int. Symposium on Algorithms and Computation (ISAAC)*, volume 1741 of Springer LNCS, pages 113–122, 1999.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proc. Discrete Algorithms and Methods for Mobility (Dial-M)*, pages 48–55, 1999.

[5] C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in Clustered Multihop, Mobile Wireless Networks. In *Proc. IEEE Singapore Int. Conf. on Networks*, pages 197–211, 1997.

[6] S. Datta, I. Stojmenovic, and J. Wu. Internal Node and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks. In *Cluster Computing 5*, pages 169–178. Kluwer Academic Publishers, 2002.

[7] P.G. Doyle and J.L. Snell. *Random Walks and Electric Networks*. The Carus Mathematical Monographs. The Mathematical Association of America, 1984.

[8] G.G. Finn. Routing and Addressing Problems in Large Metropolitan-scale Internetworks. Technical Report ISI/RR-87-180, USC/ISI, March 1987.

[9] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete Mobile Centers. In *Proc. 17^{th} Annual Symposium on Computational Geometry (SCG)*, pages 188–196. ACM Press, 2001.

[10] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric Spanner for Routing in Mobile Networks. In *Proc. ACM Int. Symposium on Mobile ad hoc networking & computing (MobiHoc)*, 2001.

[11] 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. Springer, 1996.

[12] T.C. Hou and V.O.K. Li. Transmission Range Control in Multihop Packet Radio Networks. *IEEE Transactions on Communications*, 34(1):38–44, 1986.

[13] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz, and R.E. Stearns. NC-Approximation Schemes for NP- and PSPACE-Hard Problems for Geometric Graphs. *J. Algorithms*, 26(2):238–274, 1998.

[14] D.B. Johnson and D.A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[15] B. Karp and H.T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. 6^{th} Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*, pages 243–254, 2000.

[16] Y. Ko and N. Vaidya. Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms. In *Proc. 2^{nd} IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1999.

[17] E. Kranakis, H. Singh, and J. Urrutia. Compass Routing on Geometric Networks. In *Proc. 11^{th} Canadian Conference on Computational Geometry*, pages 51–54, 1999.

[18] P. Krishna, M. Chatterjee, N. H. Vaidya, and D. K. Pradhan. A Cluster-based Approach for Routing in Ad-Hoc Networks. In *Proc. 2^{nd} USENIX Symposium on Mobile and Location-Independent Computing*, pages 1–10, 1995.

[19] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically Optimal Geometric Mobile Ad-Hoc Routing. In *Proc. 6^{th} Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial-M)*, pages 24–33. ACM Press, 2002.

[20] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. Technical Report 390, ETH Zurich, Switzerland, December 2002. http://www.inf.ethz.ch/research/publications/show.php?what=390.

[21] M. Mauve, J. Widmer, and H. Hartenstein. A Survey on Position-Based Routing in Mobile Ad-Hoc Networks. *IEEE Network*, 15(6), 2001.

[22] J.C. Navas and T. Imielinski. GeoCast - Geographic Addressing and Routing. In *Proc. Int. Conf. on Mobile Computing and Networking (MobiCom)*, pages 66–76, 1997.

[23] H. Takagi and L. Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[24] 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.

[25] Y. Wang and X.-Y. Li. Geometric Spanners for Wireless Ad Hoc Networks. In *Proc. 22^{nd} International Conference on Distributed Computing Systems (ICDCS)*, 2002.

[26] 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.

[27] Y. Xue, B. Li, and K. Nahrstedt. A Scalable Location Management Scheme in Mobile Ad-hoc Networks. In *Proc. IEEE Conference on Local Computer Networks (LCN)*, 2001.

Appendix

A Proof of Lemma 4.1

Proof. In order to prove Lemma 4.1, 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 4). 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 4, the border can be broken down into rectangles and sectors of circles. For each side of the polygon R we obtain 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. □

A smaller constant than $8/\pi$ could be obtained by placing the disks on a hexagonal grid and considering the portion of the area which is only covered by a single disk.

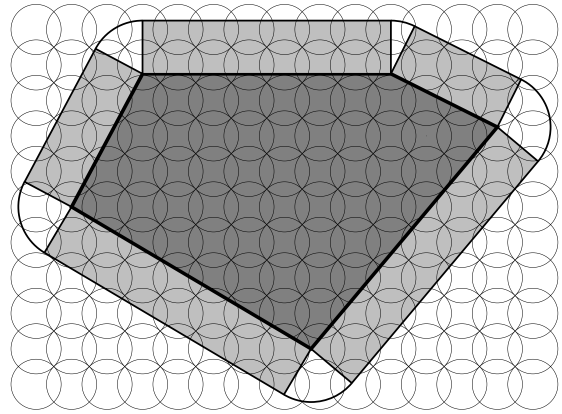

Figure 4: Covering a convex region with a grid of equally sized disks

B Proof of Lemma 4.2

Proof. We first assume there is a connection from s to t within C_i. For the definition of a *round* we distinguish three cases: According to the current algorithm execution, a *round* can be either a) a greedy step, b) a face routing phase terminated by early fallback, or c) a face routing phase terminated after exploration of the complete boundary of the current face and advancing to the node closest to t. We show that after every round the algorithm is closer to t than before that round: This holds in case a), since a greedy step can only reduce the distance to t, and in case b), as the fallback condition can only hold immediately after incrementing the counter p (that is after visiting at least one closer node) and since the algorithm then advances to the node seen so far closest to t; in case c) the algorithm approaches t, since the boundary of the currently explored face—this face contains points closer to t than where this round started—contains a point closer to t iff there is a connection to t. (Note that graphs can be constructed, where a face F's boundary contains *points* but not *nodes* that are closer to t than a given boundary node, in which case the algorithm could fail. Since we employ the Gabriel Graph, such cases can however not occur: The algorithm can forward to the a face boundary's *node* closest to t.) Since the algorithm reduces the distance to the destination with each round, it finally reaches t.

If s and t are not connected within C_i, GOAFR$^+$—in face routing mode—either hits C_i twice without finding a node closer to t (in which case the algorithm will continue on a bigger circle, which is beyond the scope of this lemma), or it explores the complete boundary of the current face (cf. above case c)) without finding a node closer to t, which is the case iff s and t are not connected at all. □

C Proof of Lemma 5.1

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

$$c(p) \leq \alpha_1 c_d(p) + \beta_1 \text{ and} \tag{3}$$

$$c(p) \geq \alpha_2 c_d(p) + \beta_2. \tag{4}$$

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 (3). 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

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

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

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

which proves Inequality (3). 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 (4), 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 consecutive 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

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

which concludes the proof. □

D Proof of Lemma 5.2

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

$$c_2(p_1^*) \geq c_2(p_2^*). \tag{5}$$

p_1^* and p_2^* are cycle free and therefore we can apply Lemma 5.1. We then obtain

$$c_2(p_1^*) \in \Theta(c_1(p_1^*)) \text{ and } c_1(p_2^*) \in \Theta(c_2(p_2^*)). \tag{6}$$

Combining Equations (5) and (6) 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. □