Hardness Results for Computing Optimal Locally Gabriel Graphs

Abhijeet Khopkar* Sathish Govindarajan†

Abstract

Delaunay and Gabriel graphs are widely studied geometric proximity structures. Motivated by applications in wireless routing, relaxed versions of these graphs known as *Locally Delaunay Graphs* (*LDGs*) and *Locally Gabriel Graphs* (*LGGs*) have been proposed. We propose another generalization of *LGGs* called *Generalized Locally Gabriel Graphs* (*GLGGs*) in the context when certain edges are forbidden in the graph. Unlike a Gabriel Graph, there is no unique *LGG* or *GLGG* for a given point set because no edge is necessarily included or excluded. This property allows us to choose an *LGG*/*GLGG* that optimizes a parameter of interest in the graph. We show that computing an edge maximum *GLGG* for a given problem instance is NP-hard and also APX-hard. We also show that computing an *LGG* on a given point set with dilation $\leq k$ is NP-hard. Finally, we give an algorithm to verify whether a given geometric graph $G = (V, E)$ is a valid *LGG*.

1 Introduction

A geometric graph $G = (V, E)$ is an embedding of the set V as points in the plane and the set E as line segments joining two points in V. Delaunay graphs, Gabriel graphs and Relative neighborhood graphs (RNGs) are classic examples of geometric graphs that have been extensively studied and have applications in computer graphics, GIS, wireless networks, sensor networks, etc (see survey [7]). Gabriel and Sokal [5] defined the Gabriel graph as follows:

Definition 1 *A geometric graph $G = (V, E)$ is called a Gabriel graph if the following condition holds: For any $u, v \in V$, an edge $(u, v) \in E$ if and only if the circle with $\overline{uv}$ as diameter does not contain any other point of V.*

Gabriel graphs have been used to model the topology in a wireless network [3]. Motivated by applications in wireless routing, Kapoor and Li [8] proposed a relaxed version of Delaunay/Gabriel graphs known as k-locally Delaunay/Gabriel graphs. The edge complexity of these structures has been studied in [8, 11]. In this paper, we focus on 1-locally Gabriel graphs and call them *Locally Gabriel Graphs* (*LGG*s).

Definition 2 *A geometric graph $G = (V, E)$ is called a Locally Gabriel Graph if for every $(u, v) \in E$, the circle with $\overline{uv}$ as diameter does not contain any neighbor of u or v in G.*

The above definition implies that in an *LGG*, two edges $(u, v) \in E$ and $(u, w) \in E$ *conflict* with each other and cannot co-exist if $\angle uwv \geq \frac{\pi}{2}$ or $\angle uvw \geq \frac{\pi}{2}$. Conversely if edges (u, v) and (u, w) co-exist in an *LGG*, then $\angle uwv < \frac{\pi}{2}$ and $\angle uvw < \frac{\pi}{2}$. We call this condition an *LGG constraint.*

Study of these graphs was initially motivated by design of dynamic routing protocols for *ad hoc* wireless networks [10]. Like Gabriel Graphs, *LGGs* are also proximity-based structures that capture the interference patterns in wireless networks. An interesting point to be noted is that there is no unique *LGG* on a given point set since no edge in an *LGG* is necessarily included or excluded. Thus the edge set of the graph (used for wireless communication) can be customized to optimize certain network parameters depending on the application. While a Gabriel graph has a linear number of edges (planar graph), an *LGG* can be constructed with a super-linear number of edges [4]. A dense network can be desirable for applications like broadcasting or multicasting. The dilation or spanning ratio of a graph is an important parameter in wireless network design. Graphs with small spanning ratios are important in many applications and motivate the study of geometric spanners. In this paper, we initiate the study of dilation on *LGGs*. We show that there exists a point set such that the Gabriel Graph on it has dilation $\Omega(\sqrt{n})$ whereas there exists an *LGG* on the same point set with dilation $O(1)$.

In many situations, certain links are forbidden in a network due to physical barriers, visibility constraints or limited transmission radius. Thus, all pairs of nodes might not induce edges and this effect can be considered in *LGGs*. Thus, it is natural to study *LGGs* in the context when the network has to be built only with a set of predefined links. In this context, we define a generalized version of *LGGs* called *Generalized locally Gabriel Graphs* (*GLGGs*). Edges in a *GLGG* can be picked only from the edges in a given predefined geometric graph.

*Dept of Computer Science and Automation, Indian Institute of Science Bangalore, abhijit@csa.iisc.ernet.in

†Dept of Computer Science and Automation, Indian Institute of Science Bangalore, gsat@csa.iisc.ernet.in

Definition 3 *For a given geometric graph $G = (V, E)$ we define $G' = (V, E')$ as GLGG if G' is a valid LGG and $E' \subseteq E$.*

Previous results on $LGGs$ have focused on obtaining combinatorial bounds on the maximum edge complexity. In [8], it was shown that an LGG has at most $O(n^{\frac{3}{2}})$ edges since $K_{2,3}$ is a forbidden subgraph. Also, it was observed in [11] that any unit distance graph is also a valid LGG. Hence there exist $LGGs$ with $\Omega(n^{1+\frac{c}{\log \log n}})$ edges [4]. It is not known whether an edge maximum LGG can be computed in polynomial time.

Our Contribution: We present the following results in this paper.

1. We show that computing a $GLGG$ with at least m edges on a given geometric graph $G = (V, E)$ is NP-complete (reduction from 3-SAT) and also APX-hard (reduction from MAX-(3,4)-SAT).
2. We show that the problem of determining whether there exists an LGG with dilation $\leq k$ is NP-hard by reduction from the partition problem motivated by [6]. We also show that there exists a point set P such that any LGG on P has dilation $\Omega(\sqrt{n})$ that matches with the best known upper bound [2].
3. For a given geometric graph $G = (V, E)$, we give an algorithm with running time $O(|E| \log |V| + |V|)$ to verify whether G is a valid LGG.

2 Hardness of computing an edge maximum $GLGG$

In this section we show that deciding whether there exists a $GLGG$ on a given geometric graph $G = (V, E)$ with at least m edges for a given value of m is NP-complete by a reduction from 3-SAT. We further show that computing edge maximum $GLGG$ is APX-hard by showing a reduction from MAX-(3,4)-SAT.

A 3-SAT instance is a conjunction of several clauses and each clause is a disjunction of exactly 3 variables. Let $\mathcal{I}$ be an instance of the 3-SAT problem with k clauses $C_1, C_2, \ldots, C_k$ and n variables $y_1, y_2, \ldots, y_n$. A geometric graph $G = (V, E)$ is constructed from $\mathcal{I}$ such that there exists a $GLGG$ on G with at least m edges if and only if $\mathcal{I}$ admits a satisfying assignment. We construct a vertex set V (points in the plane) of size $(k+3)n + k$ that is partitioned into $2n$ literal vertices denoted by $V_1 = \{x_i, x'_i \mid i \in \{1, \ldots, n\}\}$, $(k+1)n$ variable vertices denoted by $V_2 = \{z_{i_j} \mid i \in \{1, \ldots, n\}, j \in \{1, \ldots, k+1\}\}$ and k clause vertices denoted by $V_3 = \{c_j \mid j \in \{1, \ldots, k\}\}$. Thus, $V = V_1 \cup V_2 \cup V_3$. Two literal vertices x_i and x'_i corresponding to the same variable are called conjugates of each other.

Now let us discuss the placement of these vertices on the plane as shown in Figure 1. All literal vertices are placed closely on a vertical line l and the distance between two consecutive vertices is 10^{-5}. Two conjugate literal vertices corresponding to the same variable are kept next to each other. Let l_1 and l_2 be two horizontal lines passing through the highest and the lowest literal vertex respectively. Let b_0 be the center point of the line segment containing all the literal vertices. With b_0 as center, a circle is drawn with radius $d_1 = n^4$. All clause vertices $c_1, c_2, \ldots, c_k$ are placed along an arc a_0 of this circle (with a distance of $\frac{n}{2}$ between two consecutive vertices) with the additional restriction that these vertices cannot lie between lines l_1 and l_2. b_0c_1 and b_0c_k make an angle less than $\alpha = \frac{\pi}{4}$ with the horizontal axis. Now $k+1$ variable vertices are placed for each variable in the 3-SAT instance. Consider two horizontal lines l_{x_i} and $l_{x'_i}$ passing through literal vertices x_i and x'_i. With center at the mid point of x_i and x'_i (call it b_i) a circle is drawn with radius $d_2 = 10n^4$. Variable vertices are placed on an arc a_i of this circle on the same side of l where clause vertices are placed. These vertices $z_{i_1}, \ldots, z_{i_{(k+1)}}$ are placed a distance of $\frac{n}{4}$ apart with the restriction that no vertex should be placed between lines l_{x_i} and $l_{x'_i}$. Any line connecting these vertices with x_i and x'_i makes an angle less than α with the horizontal axis. For all the variables in $\mathcal{I}$, corresponding variable vertices are placed similarly. For simplicity variable vertices are shown corresponding to only one variable in Figure 1. For each clause C_j, there are 3 edges between

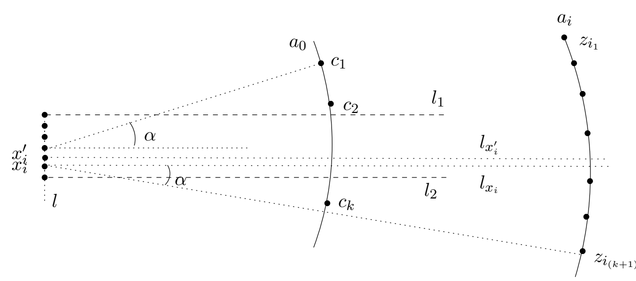

Figure 1: Placement of vertex set V

clause vertex c_j and the corresponding literal vertices. Let E_1 be the set of these edges from all the clause vertices to three corresponding literal vertices. For example, if a clause C_j has literals y_a, y_b and y'_c, then the edges $(c_j, x_a), (c_j, x_b)$ and (c_j, x'_c) are included in E_1. Another set of edges between literal vertices and variable vertices is defined

$$E_2 = \{(x_i, z_{i_1}), \ldots, (x_i, z_{i_{k+1}}), (x'_i, z_{i_1}), \ldots, (x'_i, z_{i_{k+1}}) \mid 1 \leq i \leq n\}$$

Now, $E = E_1 \cup E_2$. Let $G = (V, E)$ be the geometric graph over which an edge maximum $GLGG$ is to be computed. Let us analyze the conflicts among the edges in G. It should be noted that since a $GLGG$ is also an

LGG, it suffices to look at the *LGG* constraints to determine whether two edges conflict. Consider any *GLGG* $G' = (V, E')$ with $E' \subseteq E$ on the geometric graph G. The following constraints are observed on the edge set E'.

Since the edges (x_i, z_{i_j}) and (x'_i, z_{i_j}) conflict with each other ($\angle z_{i_j} x_i x'_i$ or $\angle z_{i_j} x'_i x_i$ is greater than $\frac{\pi}{2}$ by construction), a variable vertex z_{i_j} can have an edge incident to only x_i or x'_i.

Remark 1 *A variable vertex z_{i_j} can have only one edge ((x_i, z_{i_j}) or (x'_i, z_{i_j})) incident to it in E'.*

Similarly, we can infer Remark 2 due to *LGG* constraints.

Remark 2 *Any clause vertex c_j can be incident to at most one literal vertex in E'.*

It can be observed that two *LGG* edges that are the radii of the same circle do not conflict with each other. Here, b_i (the center of arc a_i) is close enough to both the literal vertices (x_i and x'_i) and the radius d_2 is chosen large enough so that no two edges from a literal vertex to the corresponding variable vertices conflict with each other.

Remark 3 *A literal vertex x_i (or x'_i) can have edges incident to all the corresponding variable vertices z_{i_j} in E' where $j \in \{1, \dots, k+1\}$.*

Since a literal vertex is placed sufficiently close to b_0 (the center of arc a_0) and the radius d_1 is chosen large enough, no two edges from a literal vertex to the clause vertices conflict with each other.

Remark 4 *In E', a literal vertex x_i can have edges incident to all the clause vertices that have edges incident to x_i in E_1.*

Since d_2 is chosen large enough compared to d_1, if a literal vertex x_i is connected to a variable vertex z_{i_j}, the circle with $\overline{x_i z_{i_j}}$ as diameter would contain all the clause vertices. Therefore, x_i cannot be connected to any clause vertex due to the *LGG* constraint.

Remark 5 *In E', if a literal vertex has an edge incident to a variable vertex, it cannot have an edge incident to any clause vertex.*

Lemma 1 *If there exists a GLGG G' on G with at least $(k+1)n + k$ edges, then there exists a satisfying assignment to the given 3-SAT instance.*

Proof. Since each variable vertex can have at most one edge incident to it (refer to Remark 1), at most $(k+1)n$ edges of E' can be selected from E_2. Similarly each clause vertex can have at most one edge incident to it (refer to Remark 2), so in E' at most k edges can be selected from E_1. If there are $(k+1)n + k$ edges in E', then one edge is incident to each variable vertex and clause vertex. If there is an edge between a clause vertex c_j and the literal vertex x_i (resp. x'_i), assign $y_i = 1$ (resp. $y_i = 0$) as it satisfies the clause C_j. By this rule assign a truth value to a variable in each clause. If one clause vertex is incident to x_i, no other clause vertex can be incident to x'_i as x'_i is connected to the corresponding $k+1$ variable vertices (refer to Remark 5). Therefore, this rule would yield a consistent assignment satisfying all the clauses. Hence, the given 3-SAT instance is satisfiable. □

Lemma 2 *If there is a satisfying assignment to the given 3-SAT instance, then there exists a GLGG G' over G with at least $(k+1)n + k$ edges.*

Proof. A *GLGG* with $(k+1)n + k$ edges can be constructed based on the satisfying assignment to the given 3-SAT instance. If a variable $y_i = 1$ (resp. $y_i = 0$) then connect x'_i (resp. x_i) to the corresponding $k+1$ variable vertices $(z_{i_1}, z_{i_2}, \dots, z_{i_{k+1}})$. Applying this rule to each variable we get $(k+1)n$ edges in E' from E_2 and these edges do not conflict with each other (refer to Remark 3). Since all the clauses will have at least one literal satisfied in this assignment, every clause vertex can have an edge incident to some literal vertex that has no edges incident to any of the variable vertices. Consider a clause C_j which is satisfied by the assignment $y_i = 1$ (resp. $y_i = 0$). Add the edge (c_j, x_i) (resp. (c_j, x'_i)) to E'. Since all the clauses are satisfied, k edges from E_1 can be added to E'. Therefore, G' has $(k+1)n + k$ edges and it is a valid *GLGG*. □

Theorem 3 *Deciding whether there exists a GLGG with at least m edges for a given value of m is NP-complete.*

Proof. By Lemma 1 and Lemma 2, this problem is NP-hard. Given a geometric graph G', it can be verified in polynomial time whether G' is a valid *GLGG* with at least m edges. Thus, this problem is NP-complete. □

This reduction to argue NP-hardness can be extended further to show inapproximability for computing an edge maximum *GLGG*. Let us consider the optimization version of 3-SAT known as MAX-3-SAT. Here the objective is to find a binary assignment satisfying the maximum number of clauses. MAX-(3,4)-SAT is a special case of MAX-3-SAT with an additional restriction that a variable is present in exactly four clauses. MAX-(3,4)-SAT is shown to be *APX*-hard in [1].

Now we enhance our existing construction such that for each variable there are 5 variable vertices instead of $k+1$ as described in the previous reduction. Let $G = (V, E)$ be this new geometric graph on which an optimal *GLGG* has to be computed. Again edge sets

E_1 and E_2 are defined as earlier. Now, we present the following lemma that helps to prove that computing an edge maximum $GLGG$ is APX-hard.

Lemma 4 *If a GLGG G'_1 computed over G has less than $5n$ edges from E_2 then we can obtain another GLGG G'_2 over G with $5n$ edges from E_2 and $|E'_2| \geq |E'_1|$.*

Proof. Initially let $G'_2 = G'_1$. In G'_2 if a variable vertex z_{i_j}, $1 \leq j \leq 5$ has an edge incident to an associated literal vertex x_i, then x_i cannot have an edge incident to a clause vertex (refer to Remark 5). Now x_i can have edges incident to all the five variable vertices (refer to Remark 3). Therefore, if a variable vertex z_{i_j} has an edge incident to x_i and some other variable vertex $z_{i_{j'}}$ corresponding to the same variable has no edge incident to it, then an edge $(x_i, z_{i_{j'}})$ can be added to E'_2 without conflicting with any existing edge.

If no vertex z_{i_j}, $1 \leq j \leq 5$ has an edge incident to x_i, the solution can be improved locally. Add the edges $\{(x_i, z_{i_j}) | 1 \leq j \leq 5\}$ to E'_2 and remove any edges connecting x_i to the clause vertices from E'_2. Note that a variable occurs only in four clauses in a MAX-(3,4)-SAT instance, so a literal vertex cannot have edges incident to more than four clause vertices. Therefore, this transformation implies $|E'_2| \geq |E'_1|$. Applying this argument to all the variable vertices, it can be ensured that in G'_2 every variable vertex has an edge incident to it. Thus, E'_2 has $5n$ edges from E_2 and $|E'_2| \geq |E'_1|$. $\square$

Theorem 5 *Computing an edge maximum GLGG on a given geometric graph $G = (V, E)$ is APX-hard.*

Proof. Let OPT_G and OPT_S denote the optimum for the $GLGG$ instance and the MAX-(3,4)-SAT instance respectively. A clause vertex can have only one edge incident to it (refer to Remark 2) and a $GLGG$ maximizing the edges will have $5n$ edges from E_2 (edges between variables vertices and literal vertices, refer to Lemma 4). Therefore, $OPT_G = 5n + OPT_S$. Let an algorithm maximizing the number of edges selects m edges from E_1 (edges between clause vertices and literal vertices) along with $5n$ edges from E_2. Each of these m edges implies a satisfied clause in the original MAX-(3,4)-SAT instance. Since MAX-(3,4)-SAT cannot be approximated beyond 0.99948 [1], $m < 0.99948 * OPT_S$. Let c be the best approximation bound for the edge maximum $GLGG$. Therefore, $c \leq \frac{5n+0.99948*OPT_S}{5n+OPT_S}$. Since any binary assignment or its complement would necessarily satisfy at least half of the clauses in any given 3-SAT formula, $OPT_S \geq \frac{k}{2}$. Here $n = \frac{3}{4}k$ implying $c \leq 0.999939$. Thus, it is NP-hard to approximate edge maximum $GLGG$ within a factor of 0.999939. $\square$

Consider the *maximum weight LGG* problem where the edges are assigned weights and we have to compute an LGG maximizing the sum of the weights of the selected edges. The edge maximum $GLGG$ problem is a special case of the maximum weight LGG problem (edge weights are either 0 or 1).

Corollary 1 *Computing a maximum weight LGG is APX-hard.*

3 Dilation of LGG

Let us define dilation of a geometric graph $G = (V, E)$. Let $D_G(u, v)$ be the distance between two vertices in the geometric graph (sum of length of the edges in the shortest path) and $D_2(u, v)$ be the Euclidean distance between u and v. Let $\delta(u, v) = \frac{D_G(u,v)}{D_2(u,v)}$. The dilation of G is defined as $\delta(G) = \max_{u,v \in V, u \neq v} \delta(u, v)$. In this section, we focus on computational and combinatorial questions on dilation for $LGGs$.

3.1 Computation of a minimum dilation LGG

In this section we show that the problem of determining whether there exists an LGG on a given point set with dilation ≤ 7 is NP-hard. The reduction from the partition problem is motivated by a technique in [6], where it was shown that computing the minimum dilation geometric graph with bounded number of edges is NP-hard. Since our problem requires us to construct an LGG instead of any geometric graph with bounded number of edges, the construction needs to be substantially modified.

The partition problem is defined as follows: Given a set S of positive integers $r_i, 1 \leq i \leq s$ s.t. $\sum_{r \in S} r = 2R$, can it be partitioned into two disjoint sets S_1 and S_2 such that $\sum_{r \in S_1} r = \sum_{r \in S_2} r = R$? Given an instance of the partition problem, we construct a point set V such that the instance of the partition problem is a *yes* instance if and only if there exists an LGG on V with dilation ≤ 7. Define a parameter λ s.t. $2sr^2_{max} < 10^{\lambda}$ where r_{max} is the largest element of S. For each $r_i \in S$, there is a gadget G_i. Define a parameter $\eta_i = 10^{-(\lambda+1)} r_i$ to be used in gadget G_i. Note that $\eta_i \leq \frac{1}{10}$.

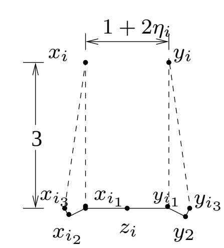

Figure 2: Structure of a basic gadget

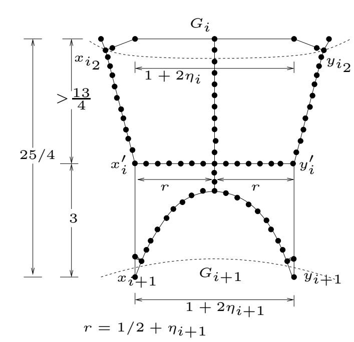

Figure 3: Basic frame structure

Now we explain the structure of a gadget G_i. Each gadget comprises of 9 points as shown in Figure 2. Points x_i and y_i are placed a distance of $1+2\eta_i$ apart. x_{i_1} and y_{i_1} are placed at the same distance such that $\overline{x_i x_{i_1}}$ and $\overline{y_i y_{i_1}}$ are parallel to each other and perpendicular to $\overline{x_{i_1} y_{i_1}}$. Vertex z_i is placed at the midpoint of the line segment $\overline{x_{i_1} y_{i_1}}$. $\overline{x_{i_1} x_{i_3}}$ is perpendicular to $\overline{x_i x_{i_1}}$ and the distance of x_{i_1} from x_{i_3} is $10\eta_i$. For $\epsilon_1 = \frac{10^{-3}}{s^2 10^{2\lambda}}$, x_{i_2} and x_{i_3} are placed at a distance of $c_1\epsilon_1$ along x-axis and $c_2\epsilon_1$ along y-axis for suitable constants c_1 and c_2, s.t. $\angle x_{i_1} x_{i_2} x_{i_3} \geq \frac{\pi}{2}$. Vertices y_{i_2} and y_{i_3} are placed similarly. We call edges $(x_{i_3}, x_{i_2}), (x_{i_2}, x_{i_1}), (x_{i_1}, z_i), (z_i, y_{i_1}), (y_{i_1}, y_{i_2})$ and (y_{i_2}, y_{i_3}) *basic edges*. It can be verified that an *LGG* over the vertices of a gadget must contain all the basic edges to keep dilation bounded by 7. It can be observed that any other edge will conflict with at least one basic edge with the exception that the point x_i can be connected to y_i, x_{i_1} or x_{i_3} and similarly y_i can be connected to x_i, y_{i_1} or y_{i_3}. Edges (x_i, x_{i_1}) and (y_i, y_{i_1}) are called *vertical edges* while (x_i, x_{i_3}) and (y_i, y_{i_3}) are called *slanted edges*. Note that the vertical edge and the slanted edge emerging from the same point x_i or y_i conflict with each other in an *LGG*. Additional points to be described later will ensure that there cannot exist a direct edge between x_i and y_i. Though both vertices x_i and y_i can have independently either a vertical or a slanted edge incident to them, if both vertices have slanted edges then $\delta(x_i, y_i) > 7$.

Remark 6 *In a gadget G_i, there can be only one slanted edge if $\delta(x_i, y_i) \leq 7$.*

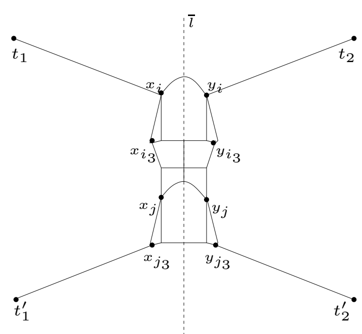

Figure 4: Layout of complete structure for $s = 2$

A frame F_i is used to connect two gadgets G_i and G_{i+1} as shown in Figure 3. It connects G_i at vertices x_{i2} and y_{i2} and connects G_{i+1} at vertices x_{i+1} and y_{i+1}. A frame also provides two symmetric paths $((x_{i+1}, x_i', x_{i_2})$ and $(y_{i+1}, y_i', y_{i_2}))$ between two consecutive gadgets. Let us denote this path length between i^{th} and $i+1^{th}$ gadget as $p_{i,i+1}$. All edges shown in the figure are part of the basic skeleton of a frame and these edges are included in the set of basic edges. Here we use a technique of placing vertices at very short distance (0.01 in our construction) from each other along a line. The purpose of this technique is to ensure that all these small edges are selected in the *LGG*. If such an edge is not selected then any alternate path does not bound the spanning ratio within limit. We call this technique *vertex closing*. It will ensure that in a frame, edges are taken only according to our layout. Such a sequence of vertices is called a *vertex chain*. An additional *auxiliary vertex* is placed in each gadget G_i at a distance of $\frac{\epsilon_1 \eta_i}{10s}$ from x_{i2} and y_{i2} along the lines $\overline{x_{i2} x_i'}, \overline{x_{i_2} x_{i_1}}, \overline{y_{i_2} y_i'}$ and $\overline{y_{i_2} y_{i_1}}$.

A frame also provides a convex cap on (x_i, y_i) in a gadget G_i. This is a convex point set with all the points above $\overline{x_i y_i}$ (it need not be a regular curve). There is a small edge incident to both x_i and y_i from this cap conflicting with the edge (x_i, y_i) and it ensures that x_i and y_i are not directly connected by an edge. It provides a path between x_i and y_i with spanning ratio just above 7 and for any other pair of vertices in it spanning ratio is bounded by 7. On the first gadget G_1, such a cap is placed explicitly as shown in Figure 4. Now the full structure is constructed as shown in Figure 4. There is a central vertical line $\bar{l}$ and all the gadgets are placed along it keeping vertex z of a gadget on $\bar{l}$ s.t. $\overline{x_{i_1} y_{i_1}}$ is perpendicular to $\bar{l}$ and a frame is placed between two gadgets. The vertical span for a frame F_i is $\frac{25}{4}$. There is a total of four extended arms, each of length h with *vertex closing* from G_1 and G_s, each making an angle $\sin^{-1}(\frac{220}{221})$ w.r.t. $\bar{l}$ (refer Figure 4). Here,

$$h = \frac{221}{148}(18s + (s-1)\frac{175}{4}) - \frac{k}{2} + \frac{1}{2}10^{-\lambda}R - \frac{1}{2}10^{-2\lambda}sr_{max}^2$$

where $k = \sum_{i=1}^{s-1} p_{i.i+1} + 10\sum_{i=1}^{s} \eta_i$.

Let V be the set of all points introduced above. Clearly $|V| = O(s)$. It can be verified that the description complexity of point set V is polynomial in the size of the partition instance.

Lemma 6 *If the partition problem S is solvable then there exists an LGG on V with dilation not exceeding 7.*

Lemma 7 *If there exists an LGG on V with dilation less than or equal to 7 then there exists a solution for the partition problem over S.*

Refer to full version [9] for the proofs of Lemma 6 and Lemma 7.

Theorem 8 *Given a point set P, it is NP-hard to find whether there exists an LGG with dilation less than or equal to a given value k.*

Proof. The proof can be inferred by Lemma 6 and Lemma 7. $\square$

Let us present some simple combinatorial bounds on the dilation of *LGGs*.

Lemma 9 *There exists a point set P such that any LGG on P has dilation $\Omega(\sqrt{n})$.*

Lemma 10 *There exists a point set P such that the Gabriel Graph on P has dilation $\Omega(\sqrt{n})$ whereas there exists an LGG on P with dilation $O(1)$.*

Refer to full version [9] for the proofs of Lemma 9 and Lemma 10.

4 Verification Algorithm for *LGG*

Given a geometric graph $G = (V, E)$, let us consider the problem of deciding whether G is a valid *LGG*. It has to be verified that no two edges conflict with each other.

For any $u \in V$, let $\mathcal{L}_u$ be a circular list storing all neighbors of vertex u in counterclockwise order. G is a valid *LGG* if edges from a vertex u to any two consecutive members in $\mathcal{L}_u$ do not conflict with each other $\forall u \in V$. This claim follows directly from the Lemma stated below.

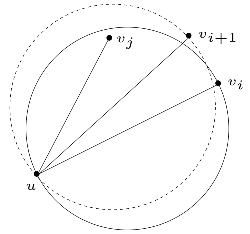

Figure 5: Checking for conflicts in an *LGG*

Lemma 11 *Let u be any vertex in G and $\mathcal{L}_u = \{v_1, v_2, \ldots, v_l\}$. If edges (u, v_i) and (u, v_j) conflict with each other such that $i \leq j-2$, then there exist a k such that $i \leq k \leq j-1$ and the edge (u, v_k) conflicts with the edge (u, v_{k+1}).*

Proof. We give a proof by contradiction. Assume that the edges (u, v_i) and (u, v_j) conflict with each other and (u, v_k) does not conflict with (u, v_{k+1}) for any k, s.t $i \leq k < j$. Let us assume w.l.o.g. that (u, v_i) and (u, v_j) are the closest pair of conflicting and non-successive edges s.t. $i \leq j-2$, i.e. if two edges (u, v'_i) and (u, v'_j) conflict with each other and $i \leq i' < j' \leq j$ then $j' = i' + 1$. Since (u, v_i) and (u, v_j) conflict with each other, let us assume w.l.o.g. that v_j lies within the circle with diameter $\overline{uv_i}$ as shown in Figure 5. By assumption (u, v_i) and (u, v_{i+1}) do not conflict, so v_{i+1} must lie outside this circle and similarly v_i will lie outside the circle with diameter $\overline{uv_{i+1}}$. Recall that two circles can intersect only at two points. Now it can be trivially observed that the circle with diameter $\overline{uv_{i+1}}$ will contain v_j. Thus, (u, v_{i+1}) and (u, v_j) conflict with each other. This implies that either (u, v_i) and (u, v_j) are not the closest pair of conflicting edges or (u, v_{i+1}) and (u, v_j) are successive edges and they do conflict with each other. In either case we have a contradiction of the original assumption. □

The argument above directly implies a verification algorithm for *LGG*. It involves computing $\mathcal{L}_u, \forall u \in V$ that can be done by angular sorting of the neighbors of each vertex. It can be implemented in $O(|E| \log |V|)$ time. Scanning each vertex u and verifying that edges to two consecutive members in $\mathcal{L}_u$ do not conflict takes $O(|V| + |E|)$ time. Therefore, this algorithm has time complexity of $O(|E| \log |V| + |V|)$.

References

[1] P. Berman, M. Karpinski, and A. D. Scott. Approximation hardness and satisfiability of bounded occurrence instances of SAT. *Electronic Colloquium on Computational Complexity (ECCC)*, 10(022), 2003.

[2] P. Bose, L. Devroye, W. S. Evans, and D. G. Kirkpatrick. On the spanning ratio of Gabriel graphs and beta-skeletons. *SIAM J. Discrete Math.*, 20(2):412–427, 2006.

[3] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 48–55, 1999.

[4] P. Erdős. On sets of distances of n points. *The American Mathematical Monthly*, 53(5):pp. 248–250, 1946.

[5] R. K. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, September 1969.

[6] P. Giannopoulos, R. Klein, C. Knauer, M. Kutz, and D. Marx. Computing geometric minimum-dilation graphs is NP-hard. *Int. J. Comput. Geometry Appl.*, 20(2):147–173, 2010.

[7] J. Jaromczyk and G. Toussaint. Relative neighborhood graphs and their relatives. *P-IEEE*, 80:1502–1517, 1992.

[8] S. Kapoor and X.-Y. Li. Proximity structures for geometric graphs. In *International Journal of Computational Geometry and Applications*, volume 20, pages 415–429, 2010.

[9] A. Khopkar and S. Govindarajan. On computing optimal locally Gabriel graphs. *CoRR*, abs/1110.1180, 2011.

[10] X.-Y. Li, G. Calinescu, and P.-J. Wan. Distributed construction of a planar spanner and routing for ad hoc wireless networks. In *IEEE INFOCOM*, pages 1268–1277, 2002.

[11] R. Pinchasi and S. Smorodinsky. On locally Delaunay geometric graphs. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 378–382, New York, NY, USA, 2004. ACM.