Robust Algorithms for Finding Triangles and Computing the Girth in Unit Disk and Transmission Graphs

Katharina Klost[1] and Wolfgang Mulzer[1]

1 **Institut für Informatik, Freie Universität Berlin**
{kathklost, mulzer}@inf.fu-berlin.de

Abstract

We describe optimal *robust* algorithms for finding a triangle and the unweighted girth in a unit disk graph, as well as finding a triangle in a transmission graph. In the robust setting, the input is not given as a set of sites in the plane, but rather as an abstract graph. The input may or may not be realizable as a unit disk graph or a transmission graph. If the graph is realizable, the algorithm is guaranteed to give the correct answer. If not, the algorithm will either give a correct answer or correctly state that the input is not of the required type.

1 Introduction

Suppose we are given a set $S \subseteq \mathbb{R}^2$ of n *sites* in the plane, where each site $s \in S$ has an *associated radius* $r_s > 0$. The *disk graph* $D(S)$ on S is defined as $D(S) = (S, E)$, where $E = \{st \mid \|st\| \leq r_s + r_t\}$, with $\|st\|$ being the Euclidean distance between the sites s and t. If all associated radii are 1, $D(S)$ is called a *unit disk graph*. The *transmission graph* on S is the *directed* graph with vertex set S and a directed edge st from site s to site t if and only if $\|st\| \leq r_s$, i.e., if and only if the t lies inside the disk of radius r_s centered at s. If all radii are equal, the edges st and ts are always either both present or both absent, for any two sites $s, t \in S$, and the resulting transmission graph is equivalent to a unit disk graph. Thus, for transmission graphs, the interesting case is that the associated radii are not all the same.

There is a large body of literature on (unit) disk graphs, see, e.g., [3,4,6,9,13]. Transmission graphs are not as widely studied, but recently they have received some attention [13, 14]. Even though disk graphs and transmission graphs may have up to $\Omega(n^2)$ edges, they can be described succinctly with $O(n)$ numbers, namely the coordinates of the sites and the associated radii. Thus, the underlying geometry often makes it possible to find efficient algorithms whose running time depends only on n.

In the setting where a unit disk graph or a transmission graph is given as an abstract graph, not much is known. One possible explanation is that the problem of deciding whether an abstract graph is a (unit) disk graph or a transmission graph is $\exists\mathbb{R}$-hard [12, 15]. In fact, Kang and Müller [12] show that there are unit disk graphs whose coordinates need an exponential number of bits in their representation, so even if it is known that the input is a unit disk graph, it is not clear that a realization of the graph can be efficiently computed. As transmission graphs with unit radii are equivalent to unit disk graphs, the result carries over to transmission graphs.

Raghavan and Spinrad [19] introduced a notion of *robust* algorithms in *restricted domains*. A restricted domain is a subset of the possible inputs. In our case, it will be the domain of unit disk graphs or transmission graphs as a subdomain of all abstract graphs. Contrary to the *promise* setting, in which the algorithm only gives guarantees for inputs from the restricted domain, the output in the robust setting must always be useful. If the input comes

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

from the restricted domain, the algorithm must always return a correct result. If the input is not from the restricted domain, the algorithm may either return a correct result, or correctly state that the input does not meet the requirement. Raghavan and Spinrad [19] give a robust polynomial-time algorithm for the CLIQUE-problem in unit disk graphs.

The problem of finding a triangle or of computing the *girth* (the shortest unweighted cycle) in a graph is a basic algorithmic question in graph theory. The best know algorithm for general graphs uses matrix multiplication and runs in either $O(n^{\omega})$ or $O(n^{2\omega/(\omega+1)})$ time, where $\omega \leq 2.371552$ is the matrix multiplication constant [2, 8, 11, 20]. The best combinatorial algorithm needs $O\left(n^3/2^{\Omega(\sqrt[7]{\log n})}\right)$ time [1, 21].

For special graph classes, better results are known. In the case of *planar* graphs, Itai and Rodeh [11], and, independently, Papadimitriou and Yannakakis [17] show that a triangle can be found in $O(n)$ time, if it exists. Chang and Lu [5] give an $O(n)$ time algorithm for computing the unweighted girth in an undirected planar graph. Kaplan et al. [13] show that in the geometric setting, finding a triangle and computing the unweighted girth can be done in $O(n \log n)$ time for general disk graphs and in $O(n \log n)$ expected time for transmission graphs. They also give algorithms with the same expected running time, for finding the smallest weighted triangle in disk graphs and transmission graphs, as well as for computing the weighted girth of a disk graph. In the geometric setting, there are $\Omega(n \log n)$ lower bounds for finding (short) triangles and computing the (weighted) girth in the algebraic decision tree model [16, 18].

In this paper, we show that there are $O(n)$ time algorithms for finding a triangle and computing the girth in unit disk graph, in the robust setting. Furthermore, we extend the ideas to an algorithm for finding a triangle in a transmission graph in $O(n + m)$ time. The running times for the algorithms in the unit disk graph setting can be sublinear in the input size, as the input in the robust setting consists of a representation of all vertices and edges. In particular, the result is better than the $\Omega(n \log n)$ lower bound for the geometric setting, because this lower bound stems from the difficulty of finding the edges of the graph. The running time for transmission graphs is linear in the input size, and it is significantly faster than the currently fastest algorithm for general graphs [12].

2 Preliminaries

We assume that the input is an abstract unweighted graph $G = (V, E)$, given as an adjacency list. For an undirected graph, given a vertex v, we denote by $N(v)$ the set of vertices that are adjacent to v, and by $\deg(v) = |N(v)|$ the degree of v. In the adjacency list representation, a set of k neighbors of v can be reported in $O(k)$ time, and testing if two vertices u and v are adjacent takes $O(\min(\deg(v), \deg(u)))$ time. For a directed graph, let $N_{\text{in}}(v)$ and $N_{\text{out}}(v)$ be the vertices connected by an incoming or outgoing edge to a vertex v, respectively. Let $N_{\text{bi}} = N_{\text{in}} \cap N_{\text{out}}$ be the vertices connected by both an incoming and an outgoing edge.

▶ **Definition 2.1** (Raghavan and Spinrad [19]). A *robust* algorithm for a problem P on a domain C solves P correctly, if the input is from C. If the input is not in C, the algorithm may either produce a correct answer for P or report that the input is not in C.

3 Unit disk graphs

Our algorithms make use two key properties. Both properties are well known. For completeness, we include proofs for the variants we use.

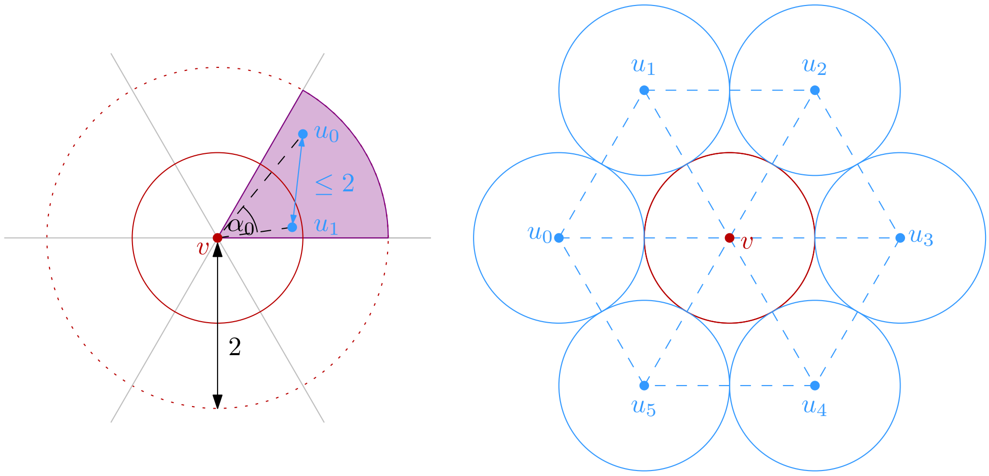

Figure 1 Left: The disk defined by v is marked in solid red. There are two sites in the shaded sector. The diameter of the sector is 2, so u_0 and u_1 are connected by an edge.
Right: The mutual distance between the u_i is maximized on the vertices of a regular polygon. The side length of each equilateral triangle is 2, and thus the desired edges exist.

▶ **Lemma 3.1.** *Let $G = (V, E)$ be a graph that is realizable as an unit disk graph, and let $v \in V$ be a vertex with $\deg(v) > 5$. Then, the subgraph induced by v and any six adjacent vertices contains a triangle.*

Proof. Consider a realization of the unit disk graph in the plane. We identify the vertices with the corresponding sites. Let $u_0, \ldots, u_5 \in N(v)$ be any six neighbors of v, labelled in clockwise order around v.

Let $\alpha_i = \sphericalangle u_i v u_{i+1}$, $i = 0, \ldots 5$, be the angles between the consecutive neighbors with respect to v, where the indices are taken modulo 6. Note that $\sum_{i=0}^{5} \alpha_i = 2\pi$, and suppose that α_0 is a minimum angle in this sequence.

First, suppose that $\alpha_0 < \pi/3$. Then, there is a cone with opening angle $\pi/3$ and apex v that contains the sites u_0 and u_1. Since u_0 and u_1 are adjacent to v, we have $\|u_0 v\| \leq 2$ and $\|u_1 v\| \leq 2$, so u_0 and u_1 both lie inside a circular sector with angle $\pi/3$, apex v, and radius 2. This circular sector has diameter 2. Thus, all points in it, in particular u_0 and u_1, have mutual distance at most 2. If follows that u_0 and u_1 are connected by an edge, closing a triangle, see Figure 1, left.

Second, suppose that $\alpha_0 \geq \pi/3$. Then, since α_0 is minimum and since the α_i sum to 2π, it follows that $\alpha_i = \pi/3$, for $i = 0, \ldots, 5$, so the u_i lie on six concentric, uniformly spaced rays that emanate from v. In this case, the maximum possible mutual distance between the u_i's is achieved when the sites constitute the corners of a regular hexagon with center v and $\|u_i v\| = 2$, for $i = 0, \ldots, 5$. This hexagon decomposes into six equilateral triangles of side length 2, and thus the unit disk graph contains all consecutive edges $\{u_i, u_{i+1}\}$, closing a triangle, see Figure 1, right. ◀

▶ **Lemma 3.2.** *If a (unit) disk graph does not contain a triangle, it is planar.*

Proof. Given the geometric representation of a unit disk graph, the natural embedding of this unit disk graph into the plane is by connecting the sites that represent the vertices by line segment. Evans et al. [7], as well as Kaplan et al. [13] show that if this embedding is not plane, there has to be a triangle in the unit disk graph. Conversely, this implies that if there is no triangle in the graph, the natural embedding is crossing free, directly implying that the graph is planar. ◀

3.1 Finding a Triangle

▶ **Theorem 3.3.** *There is a robust algorithm to find a triangle in a disk graph in $O(n)$ time.*

Proof. The algorithm works as follows. If there is no vertex v with $\deg(v) > 5$, then we check explicitly for every vertex whether two of its neighbors are adjacent. If so, we have found a triangle. If not, there is none. As all degrees are constant, this takes $O(1)$ time per vertex, for a total of $O(n)$ time.

Now, assume there is a vertex v with $\deg(v) > 5$. Let $N'(v)$ be a set of any seven neighbors of v. For every pair of neighbors u, w from $N'(v)$, explicitly check if there is an edge between u and w. If an edge is found, report the triangle u, v, w. Otherwise, report that the input is not a unit disk graph. This step takes $O(n)$ time to identify v and then at most $O(n)$ time to check the adjacencies for each of the $O(1)$ vertices in $N'(v)$, summing up to $O(n)$ total time. Note that in the case that not all degrees are at most five, only one vertex is considered in detail.

To see that the algorithm is correct, we consider all possible cases. If the maximum degree of the graph is at most 5, all vertices and their neighbors are explicitly checked. So if there is a triangle in the graph, the algorithm will find it and correctly report it. Furthermore, no triangles can be missed. Otherwise, there is a vertex v with degree larger than 5. If the input is a disk graph, Lemma 3.1 guarantees that there is a triangle in $N'(v)$. The algorithm explicitly searches for such edge between vertices of $N(v')$. If such an edge is found, the triangle is correctly reported. In the other case, Lemma 3.1 implies that the input is not a unit disk graph, as reported by the algorithm. ◀

3.2 Computing the Girth

▶ **Theorem 3.4.** *There is a robust algorithm to compute the girth of a graph in the domain of unit disk graphs that runs in $O(n)$ time.*

Proof. First, run the algorithm from Theorem 3.3 on the input. If the algorithm determines that the input graph is not a unit disk graph, report this and finish. If the algorithm found a triangle, the girth of the graph is three and can be reported.

If the algorithm from Theorem 3.3 did not find a triangle and did not report that the graph is not a unit disk graph, we use a linear time planarity testing algorithm on the graph, e.g., the algorithm described by Hopcroft and Tarjan [10]. If the graph is not planar, report that it is not a unit disk graph. In the other case, the algorithm for computing the girth in a planar graph by Chang and Lu [5] can be used to compute the girth of the graph in $O(n)$ time.

By combining the running times for each step, the overall running time follows. Correctness follows from Lemma 3.2 and the correctness of the algorithm by Chang and Lu. ◀

4 Finding a directed triangle in a transmission graph

The following key lemma needed for the robust algorithm for transmission graphs was previously shown by Klost [16]. We include a proof for completeness.

▶ **Lemma 4.1.** *If G is a transmission graph and v is a vertex with $|N_{bi}(v)| > 6$, then G contains a triangle.*

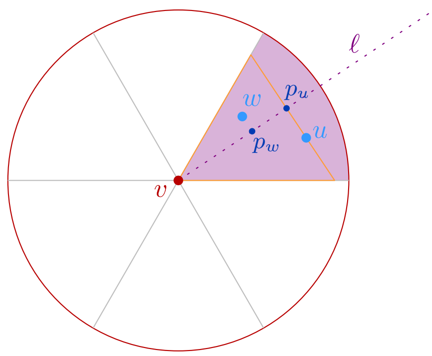

Figure 2 The disk defined by v is marked in solid red. There are two sites in the shaded sector. The distance between two points in the triangle is at most $\|uv\|$.

Proof. The proof is similar to that of Lemma 3.1. Consider a representation of G in the plane and the disk D_v associated to the site v. By definition, all sites in $N_{\mathrm{bi}}(v)$ lie in D_v. Subdivide D_v into six congruent circular sectors, as in Figure 2. By the pigeonhole principle, one sector C contains at least two vertices u and w from $N_{\mathrm{bi}}(v)$.

Let ℓ be the perpendicular bisector of C and let p_u and p_w be the perpendicular projections of u and w onto ℓ. W.l.o.g., suppose that p_w is closer to v than p_u. Consider the equilateral triangle $\triangle$ defined by the line through u perpendicular to ℓ and the rays defining C. Then, w is contained in $\triangle$, and thus $\|uw\| \leq \|uv\| \leq r_u$. Hence, the edge uw exists in the transmission graph, closing the directed triangle. ◀

▶ **Lemma 4.2.** *The sets $N_{bi}(v)$, for all $v \in V$ can be found in $O(n+m)$ time.*

Proof. We assume that the adjacency list representation has the following standard form: on the top level, there is an array that can be indexed by the vertices in V. Each entry points to the head and the tail of a linked list. This linked list contains all vertices u such that $(v,u) \in E$, in no particular order. Note that the order of the array directly induces a total order $\preceq$ on V.

We compute adjacency list representations $L_{\preceq}^{\top}$ and $L_{\preceq}$ of the transposed graph $G^{\top}$ and the original graph G, with the additional property that linked list are sorted according to $\preceq$. For this, we initialize a new array and traverse the original adjacency list representation of G, by considering the vertices according to $\preceq$. For every edge (v,u) that is encountered, we append v to the linked list of u, in $O(1)$ time. As the source vertices v are traversed according to $\preceq$, this gives the desired representation $L_{\preceq}^{\top}$ of $G^{\top}$, in time $O(n+m)$. After that, we can obtain the representation $L_{\preceq}$ of G by the same procedure, using $L_{\preceq}^{\top}$ as the initial adjacency list. Finally, to identify the sets $N_{\mathrm{bi}}(v)$, for each $v \in V$, it suffices to merge the associated lists $L_{\preceq}$ and $L_{\preceq}^{\top}$, in total time $O(n+m)$. ◀

▶ **Lemma 4.3.** *There is a robust algorithm that finds a directed triangle in a transmission graph in $O(n+m)$ time.*

Proof. Before starting the algorithm, preprocess the input as described in the proof of Lemma 4.2 in $O(n+m)$ time, such that the set $N_{\mathrm{bi}}(v)$ is known for every vertex v. If all sites have $|N_{\mathrm{bi}}(v)| \leq 6$, then for each ordered pair $(u,w) \in N_{\mathrm{bi}}(v) \times N_{\mathrm{bi}}(v)$, explicitly check if the edge uw is in the graph and if such an edge is found, report the triangle vuw. This step takes an overall of $O(n+m)$ time, as even though $|N_{\mathrm{bi}}|$ is bounded for every vertex, the same does not necessarily hold for $|N_{\text{in}}|$ and $|N_{\text{out}}|$, so potentially all edges of the graph are considered in this step.

In the other case, let v be a vertex with $|N_{\text{bi}}(v)| > 6$. Then, find a set of 7 vertices from this set and explicitly check if any pair of them closes a triangle. If there is no triangle, then report that the input is not a transmission graph. Otherwise, report the triangle. This again takes $O(n+m)$ time.

The correctness for the complete algorithm follows from the same arguments as in Theorem 3.3, using Lemma 4.1 instead of Lemma 3.1. ◀

5 Conclusion

We showed that there are robust sublinear algorithms for finding a triangle and computing the girth in unit disk graph as well as a linear time algorithm for finding a triangle in a transmission graph. Extending the arguments to general disk graphs seems to be hard, as the properties given in Lemma 3.1 and Lemma 4.1 do not easily carry over to general disk graphs. It would be interesting to see if there are properties of transmission graphs that allow a sublinear running time similar to the unit disk graph case.

References

1. Amir Abboud, Nick Fischer, Zander Kelley, Shachar Lovett, and Raghu Meka. New graph decompositions and combinatorial Boolean matrix multiplication algorithms. `arXiv:2311.09095`, 2023.
2. Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. `doi:10.1007/BF02523189`.
3. Alexander Baumann, Haim Kaplan, Katharina Klost, Kristin Knorr, Wolfgang Mulzer, Liam Roditty, and Paul Seiferth. Dynamic connectivity in disk graphs. *Discrete Comput. Geom.*, 71(1):214–277, 2024. `doi:10.1007/s00454-023-00621-x`.
4. Sergio Cabello and Miha Jejčič. Shortest paths in intersection graphs of unit disks. *Comput. Geom. Theory Appl.*, 48(4):360–367, 2015. `doi:10.1016/j.comgeo.2014.12.003`.
5. Hsien-Chih Chang and Hsueh-I Lu. Computing the girth of a planar graph in linear time. *SIAM J. Comput.*, 42(3):1077–1094, 2013. `doi:10.1137/110832033`.
6. Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1):165–177, 1990. `doi:10.1016/0012-365X(90)90358-O`.
7. William Evans, Mereke van Garderen, Maarten Löffler, and Valentin Polishchuk. Recognizing a DOG is hard, but not when it is thin and unit. In *Proc. 8th FUN*, pages 16:1–16:12, 2016. `doi:10.4230/LIPIcs.FUN.2016.16`.
8. François Le Gall. Powers of tensors and fast matrix multiplication. In *Proc. 39th International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 296–303, 2014. `doi:10.1145/2608628.2608664`.
9. A. Gräf, M. Stumpf, and G. Weißenfels. On coloring unit disk graphs. *Algorithmica*, 20(3):277–293, 1998. `doi:10.1007/PL00009196`.
10. John Hopcroft and Robert Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974. `doi:10.1145/321850.321852`.
11. Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978.
12. Ross J. Kang and Tobias Müller. Sphere and dot product representations of graphs. *Discrete Comput. Geom.*, 47(3):548–568, 2012. `doi:10.1007/s00454-012-9394-8`.

13 Haim Kaplan, Katharina Klost, Wolfgang Mulzer, Liam Roditty, Paul Seiferth, and Micha Sharir. Triangles and girth in disk graphs and transmission graphs. In *Proc. 27th Annu. European Sympos. Algorithms (ESA)*, pages 64:1–64:14, 2019. doi:10.4230/LIPIcs.ESA.2019.64.

14 Haim Kaplan, Wolfgang Mulzer, Liam Roditty, and Paul Seiferth. Spanners and reachability oracles for directed transmission graphs. In *Proc. 31st Int. Sympos. Comput. Geom. (SoCG)*, pages 156–170, 2015. doi:10.4230/LIPIcs.SOCG.2015.156.

15 Katharina Klost. Complexity of recognizing generalized transmission graphs. Master's thesis, Freie Universität Berlin, 2017.

16 Katharina Klost. *Geometric graphs: reachability, long trees and short cycles.* PhD thesis, Freie Universität Berlin, 2021.

17 Christos H. Papadimitriou and Mihalis Yannakakis. The clique problem for planar graphs. *Inform. Process. Lett.*, 13(4):131–133, 1981. doi:10.1016/0020-0190(81)90041-7.

18 Valentin Polishchuk. Personal communication. 2017.

19 Vijay Raghavan and Jeremy Spinrad. Robust algorithms for restricted domains. *J. Algorithms*, 48(1):160–172, 2003. doi:10.1016/S0196-6774(03)00048-8.

20 Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: From alpha to omega. In *Proc. 35th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 3792–3835, 2024. doi:10.1137/1.9781611977912.

21 Huacheng Yu. An improved combinatorial algorithm for Boolean matrix multiplication. *Inform. and Comput.*, 261:240–247, 2018. doi:10.1016/J.IC.2018.02.006.