Proceedings of DGA 2013, pp. 1 – 5.

A geometric trigraph model for unit disk graph recognition*

Guilherme da Fonseca[1], Vinícius Pereira de Sá[2], Raphael Machado[3], and Celina de Figueiredo[4]

[1]*Universidade Federal do Estado do Rio de Janeiro,* fonseca@uniriotec.br

[2]*DCC/IM, Universidade Federal do Rio de Janeiro,* vigusmao@dcc.ufrj.br

[3]*Inmetro — Instituto Nacional de Metrologia, Qualidade e Tecnologia,* rcmachado@inmetro.gov.br

[4]*COPPE, Universidade Federal do Rio de Janeiro,* celina@cos.ufrj.br

Abstract A unit disk graph G is a graph whose vertices can be mapped to points on the plane and whose edges are defined by pairs of points within unitary Euclidean distance from one another. The recognition of unit disk graphs is no easy feat. Indeed, the fastest known algorithm to decide whether a given graph is a unit disk graph is doubly exponential. In this paper, we introduce a practical algorithm to produce certified answers to the question "is G a unit disk graph?" in either way, for any given graph G. By imposing that the points' coordinates belong to discrete sets of increasing granularity, our method builds a sequence of trigraphs G', i.e. graphs with mandatory and optional edges, until either some G' is found possessing properties which certify that G is a unit disk graph, or the sequence of trigraphs has to be interrupted, certifying that G is not a unit disk graph. The proposed method was actually implemented, and we were able to obtain our first certificates for some small graphs.

Keywords: unit disk graphs, graph recognition, trigraphs, geometric algorithms

1. Motivation

A ***unit disk graph*** (UDG) is a graph whose n vertices can be mapped to points on the plane and whose m edges are defined by pairs of points within Euclidean distance at most 1 from one another. Alternatively, one can regard the vertices of a UDG as mapped to coplanar congruent disks, so that two vertices are adjacent whenever the corresponding disks intersect. Unit disk graphs have been widely studied in recent times due to their applications to wireless sensor networks [1].

In this paper, we consider the problem of recognizing unit disk graphs. Though a YES answer can be verified in polynomial time assuming the Real RAM model, the size of certificates comprising the coordinates of the disk centers may not be polynomially bounded under the classic model of computation over finite strings [3]. Indeed, it is not known for the time being whether the problem belongs to NP, and the fastest known recognition algorithm is doubly exponential [4]. Since no practical algorithm is available, there are graphs with as few as ten vertices which have never been proved as being (or not being) UDG [5].

A practical method to certify whether a graph is a UDG is of utmost importance. Indeed, many of the existing bounds for approximation factors of algorithms for hard problems on unit disk graphs are based on the fact that certain graphs are (or are not) UDG, but each one of

*Research partially supported by FAPERJ and CNPq.

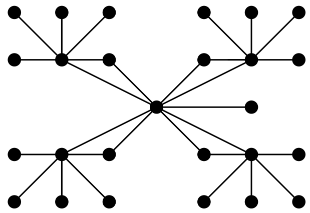

Figure 1. Graph conjectured [5] not to be a UDG.

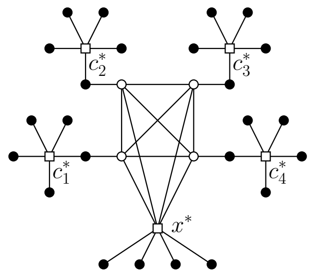

Figure 2. Graph that corresponds to the lower bound for the approximation factor of the algorithm introduced in [2] for minimum (independent) dominating sets in unit disk graphs.

those graphs demanded their own ad-hoc geometric proof. For an example, [5] conjectures that the graph in Figure 1 is not a UDG. The correctness of their conjecture would imply a decrease from 3.8 to 3.6 in the maximum ratio (except for an additive constant) between the size of a maximal independent set and the size of a connected dominating set in any given UDG, and that would immediately tighten the approximation factor of algorithms that estimate the size of minimum connected dominating sets by computing maximal independent sets.

Another example was obtained in [2]. Denote by $G_{p,q}$ the graph that has one p-clique such that one of its vertices is adjacent to q pendant vertices, and each of the other $p-1$ vertices is adjacent to a degree-2 vertex that in turn is a pendant vertex of an induced $K_{1,5}$. The graph $G_{5,4}$ of Figure 2 is known to be a UDG (a geometric model with only integral coordinates is available) and is the worst known instance for an algorithm that approximates the minimum (independent) dominating set of a unit disk graph, establishing a lower bound of 4.8 for the approximation factor of that algorithm. On the other hand, the graph $G_{9,4}$ is known not to be a UDG (the proof is based on numerous geometric lemmas), and this fact is central in the proof of the (upper bound for the) approximation factor of $44/9 = 4.888\ldots$ of such algorithm. Further knowledge about the family $G_{p,q}$, closing the gap between what is currently known to be a UDG (graph $G_{5,4}$) and what is known not to be a UDG (graph $G_{9,4}$) would immediately tighten the existing bounds on the approximation factor of the aforementioned algorithm.

The difficulty in developing a certifier for unit disk graphs, even a "brute force" one, comes from the fact that the solution space — namely $(\mathbb{R}^2)^n$ — is uncountable. In the present paper, we formulate a strategy to reduce the solution space to a countable, finite set, whose granularity is subsequently refined, leading to a YES/NO certificate in many cases. An inconclusive answer, however, may possibly be obtained.

2. The proposed model

The central idea of our strategy is to discretize the solution space by defining an enumerable set of 2-dimensional coordinates where the points associated to the input graphs' vertices may be placed at. For a positive $\epsilon \in \mathbb{R}$, consider the set $N_\epsilon := \{x \in \mathbb{R} \mid x = d\epsilon,\ d \in \mathbb{N}\}$, and let $C_\epsilon := N_\epsilon \times N_\epsilon$ be a discrete set of 2-dimensional coordinates. We call such C_ϵ a ***mesh*** and we say C_{ϵ_1} is thinner than C_{ϵ_2} if $\epsilon_1 < \epsilon_2$. Clearly, any subset of points $M_\epsilon \subseteq C_\epsilon$ determines a unit disk graph G whose vertices are pairwise adjacent whenever their corresponding points in M_ϵ are within unitary distance of one another. We say M_ϵ is an ***ϵ-discrete model*** for G.

Trigraph embodiments. Given a mesh C_ϵ and a set $M_\epsilon \subseteq C_\epsilon$ of n points, we define the trigraph $G_{M_\epsilon} = (V, E_1 \cup E_2)$ as the graph whose vertex set V corresponds to the points in M_ϵ, and whose edges can be partitioned into E_1, the set of ***mandatory*** edges, and E_2, the set of ***optional*** edges. A mandatory edge is associated to a pair of points $v, w \in M_\epsilon$ that are at distance $d(v, w) < 1 - \epsilon\sqrt{2}$ from one another. An optional edge, on its turn, is associated to a pair of points $v, w \in M_\epsilon$ satisfying $1 - \epsilon\sqrt{2} \leq d(v, w) \leq 1 + \epsilon\sqrt{2}$. We say G_{M_ϵ} is a ***trigraph embodiment*** of graph $G(V, E)$ if, and only if, $E \subseteq E_1 \cup E_2$ and $E_1 \setminus E = \emptyset$, i.e. all edges of G are either mandatory or optional edges in G_{M_ϵ}, and no edge that does not belong to G appears as a mandatory edge in G_{M_ϵ}.

If G_{M_ϵ} is a trigraph embodiment of G, and G_{M_ϵ} has no optional edges, then M_ϵ is a unit disk model for G, hence G is certainly a UDG. Moreover, if G_{M_ϵ} does have optional edges, but all optional edges in G_{M_ϵ} correspond to pairs of adjacent vertices in G, then G is a UDG as well. (The same goes for the case where all optional edges in G_{M_ϵ} correspond to pairs of non-adjacent vertices in G.) This is the core of the YES certificates produced by our method.

It can be shown that, if G is a UDG, then G admits a trigraph embodiment G_{M_ϵ}, for all $\epsilon > 0$. Conversely, if, for some ϵ, there is no possible trigraph embodiment G_{M_ϵ} for G, then G is not a UDG. Our NO certificates come from this fact.

Our strategy to recognize unit disk graphs can therefore be summarized in the following steps:

INPUT: A connected graph $G = (V, E)$
OUTPUT: YES, if G is a UDG; NO, if it is not a UDG; or INCONCLUSIVE.

1. Choose a value for ϵ and consider the corresponding mesh C_ϵ.
2. For each possible discrete model $M_\epsilon \subseteq C_\epsilon$ with $|M| = |V|$, obtain the respective trigraph $G_{M_\epsilon} = (V, E_1, E_2)$.
   (a) If $E = E_1$ then a disk model was found for G, hence G is a UDG. Return YES.
   (b) If $E \subseteq E_1 \cup E_2$ and $E_1 \setminus E = \emptyset$, then G_{M_ϵ} is a trigraph embodiment for G.
3. If a trigraph embodiment was found for G, then let $\epsilon \leftarrow \epsilon/2$. If ϵ is still greater than some previously defined constant ϵ_{min}, then restart the algorithm with the new value for ϵ; otherwise, return INCONCLUSIVE.
4. If no trigraph embodiment was found for G, then G is not a UDG. Return NO.

Note that, in spite of the apparent infinite number of possible discrete models, we may assume that G is connected[1], so any model of G must be enclosed in a disk of diameter $2|V|$.

Notice also that, whenever the algorithm produces a conclusive answer, then an appropriate certificate has been found. However, as discussed in Section 4, the input graph may not be a UDG, but still be such that, no matter how thin the mesh is, a trigraph model can always be found, leading the algorithm to an inconclusive answer.

3. Results

To validate our proposed model, we implemented it using the Python language. Our implementation includes some nice refinements aimed at reducing the number of candidate placements of each vertex in the considered mesh, such as

(i) taking the maximum and minimum distances between pairs of vertices as input;

(ii) taking the maximum and minimum angle between two vertices with respect to a third one as input;

(iii) allowing the imposition of a fixed circular order of vertices around a reference point.

Naturally, such features can only be used if some previous geometric analysis determines such distances and angles constraints. With this preliminary implementation, we could already correctly classify some small graphs as being (or not being) UDG.

[1]Trivially, a graph is a UDG if and only if all its connected components are UDG.

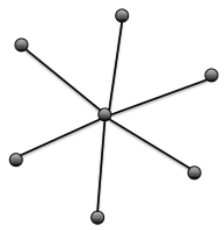

Figure 3. Graph $K_{1,6}$.

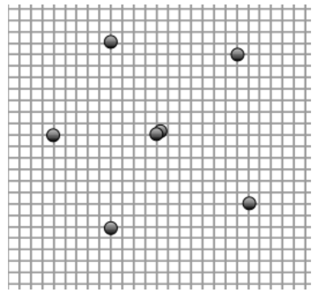

Figure 4. Discrete model for graph $K_{1,6}$. The circles are not the unit-diameter disks themselves, but rather represent their centers. The two overlapping circles represent the centers of coincident disks.

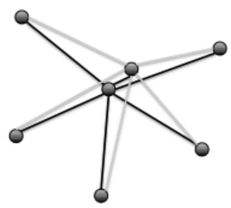

Figure 5. Trigraph corresponding to Figure 4.

4. Future directions

In spite of the nice results it has enabled us to obtain, the proposed method does presents some limitations, one of which is disclosed by the following "pathological" example.

Let G be the $K_{1,6}$ graph, which is known (by geometric methods) not to be a UDG. Our procedure is doomed to give an inconclusive answer for G no matter how thin the mesh is. The reason is that, for all $\epsilon > 0$, there is always a trigraph embodiment G_{M_ϵ} for G, in which the center of the star and one of the leaves coincide (see Figures 3, 4 and 5).

A second weakness of the method is its worst-case time complexity, since the time demanded to produce a certificate for certain graphs may be as long as unforeseeable.

The previous observations lead to the following open questions, which are currently under investigation.

1. Is it possible to characterize such "pathological" graphs, those which deny our method any chance of recognizing them in either way?

2. Is it possible to modify our method so that it always stop with a conclusive question within a reasonable, predetermined time?

Notwithstanding the open questions above, there seem to be several promising ways our method can be improved upon. We list some of them below.

- The exhaustive enumeration of possible trigraph embodiments for G can be achieved by a backtracking-based approach. First, a sequence $v_1, \ldots, v_n$ of vertices of G must be determined, in such a way that the subgraph G_k of G induced by $v_1, \ldots, v_k$ is connected for all $k \in \{1, \ldots, n\}$. Each vertex v_k is then positioned, one at a time, at some point of the mesh, in such a way that the set of already occupied points of the mesh (including the one assigned to v_k) defines a trigraph embodiment for G_k. By doing so, the search space for trigraph embodiments for G shall decrease considerably.

- By the end of the k-th iteration of the algorithm, after some trigraph embodiments were found, the value of ϵ is halved, so each former grid point p gives rise to four grid points p_1, p_2, p_3, p_4 to be considered (as possible vertex locations) during the $(k+1)$-th iteration. It shall now be possible to eliminate at once from the list of candidate locations for a vertex v all points p_i corresponding to a point p that was not occupied by v in any trigraph embodiment obtained in the k-th iteration. By so doing, the search for trigraph embodiments on the thiner mesh becomes limited to "refining" previously obtained trigraph embodiments, instead of a search that would otherwise have begun from scratch.
- Proving geometric results such as "if G is a UDG, then G admits a disk model where no two vertices are either vertically aligned, or horizontally aligned, or coincident" may allow for the earlier elimination of a considerable number of discrete models, therefore also speeding up the algorithm.

References

[1] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz (2005). Simple heuristics for unit disk graphs. Networks 25 (2): 59–68.

[2] G. D. Fonseca, C. M. H. de Figueiredo, V. G. Pereira de Sá, and R. C. S. Machado (2013). Linear Time Approximation for Dominating Sets and Independent Dominating Sets in Unit Disk Graphs, in Workshop on Approximate and Online Algorithms, Lect. Notes Comp. Sci. 7846, 82–92.

[3] C. McDiarmid, and T. Müller (2013). Integer realizations of disk and segment graphs. Journal of Combinatorial Theory, Series B 103 (1): 114–143.

[4] J. Spinrad (2003). Efficient Graph Representations. Fields Inst. monographs. AMS.

[5] W. Wu, H. Du, X. Jia, Y. Li, and S. C.-H. Huang (2006). Minimum connected dominating sets and maximal independent sets in unit disk graphs, Theor. Comput. Sci. 352(1–3), 1–7.