Computational Geometry ••• (••••) •••–•••

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Two optimization problems for unit disks ☆

Sergio Cabello [a,*], Lazar Milinković [b]

[a] FMF, University of Ljubljana, and Institute of Mathematics, Physics and Mechanics, Slovenia
[b] FMF and FRI, University of Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 10 February 2017
Received in revised form 3 October 2017
Accepted 21 December 2017
Available online xxxx

Keywords:
Shortest path tree
Unit disk graph
Geometric graph
Minimum separation
Implementation

ABSTRACT

We present an implementation of a recent algorithm to compute shortest-path trees in unit disk graphs in $O(n \log n)$ worst-case time, where n is the number of disks.
In the minimum-separation problem, we are given n unit disks and two points s and t, not contained in any of the disks, and we want to compute the minimum number of disks one needs to retain so that any curve connecting s to t intersects some of the retained disks. We present a new algorithm solving this problem in $O(n^2 \log^3 n)$ worst-case time and its implementation.

1. Introduction

In this paper we consider two geometric optimization problems in the plane where unit disks play a prominent role. For both problems we discuss efficient algorithms to solve them, provide an implementation of these algorithms, and present experimental results on the implementation.

The first problem we consider is computing a *shortest-path tree* in the (unweighted) intersection graph of unit disks. The input to the problem is a set $\mathcal{D}$ of n disks of the same size, each disk represented by its centre. The corresponding unit disk (intersection) graph has a vertex for each disk, and an edge connecting two disks D and D' of $\mathcal{D}$ whenever D and D' intersect. An alternative, more convenient point of view, is to take as vertex set the set of centres of the disks, denoted by P, and connecting two points p and q of P whenever the Euclidean length $|pq|$ is at most the diameter of a disk. The graph is unweighted. Given a root $r \in P$, the task is to compute a shortest-path tree from r in this graph. See Fig. 1.

The second problem we consider is the *minimum-separation problem*. The input is a set $\mathcal{D}$ of n unit disks in the plane and two points s and t not covered by any disks of $\mathcal{D}$. We say that $\mathcal{D}$ *separates* s and t if each curve in the plane from s to t intersects some disk of $\mathcal{D}$. The task is to find the minimum cardinality subset of $\mathcal{D}$ that separates s and t. See the left of Fig. 1 for an example of an instance. Formally, we want to solve

$$\begin{aligned} &\min \ |\mathcal{D}'| \\ &\text{s.t. } \mathcal{D}' \subseteq \mathcal{D} \text{ and } \mathcal{D}' \text{ separates } s \text{ and } t. \end{aligned}$$

☆ Supported by the Slovenian Research Agency, core program P1-0297 and project L7-5459.
* Corresponding author.
E-mail address: sergio.cabello@fmf.uni-lj.si (S. Cabello).

https://doi.org/10.1016/j.comgeo.2017.12.001
0925-7721/© 2017 Published by Elsevier B.V.

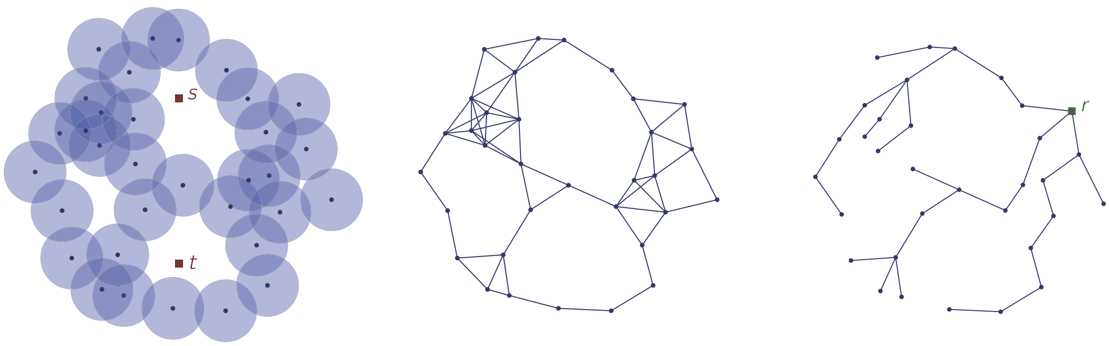

Fig. 1. Left: unit disks and two additional points s and t. Middle: intersection graph of the disks. Right: a shortest-path tree in the graph.

Unit disks are the most standard model used for wireless sensor networks; see for example [8,11,21]. Often the model is referred as UDG. This model provides an appropriate trade off between simplicity and accuracy. Other models are more accurate, as for example discussed in [14,16], but obtaining efficient algorithms for them is much more difficult.

While unit disks give a simple model, exploiting the geometric features of the model is often challenging. Shortest paths in unit disk graphs are essential for routing and are a basic subroutine for several other more complex tasks. A somehow unexpected application of shortest paths in unit-disk graphs to boundary recognition is given in [20]. The minimum-separation problem and variants thereof have been considered in [2,9]. The problem is dual to the barrier-resilience problem considered in [1,13,15]. It is not obvious that the minimum-separation problem can be solved optimally in polynomial time, and the known algorithm for this uses as a subroutine shortest paths in unit disk graphs. Thus, both problems considered in this paper are related and it is worth to consider them together.

Our contribution. We are aware of three algorithms to compute shortest-path trees in unit disk graphs in $O(n \log n)$ worst-case time: one by Cabello and Jejčič [3], one by Chan and Skrepetos [5], and one by Efrat, Itai and Katz [7]. Here we report on an implementation of a modification of the algorithm in [3], and compare it against two obvious, simple alternatives: one based on grids and one using breadth-first search after the explicit construction of the unit disk graph. The only complex ingredients in the algorithm is computing the Delaunay triangulation and static nearest-neighbour queries, but efficient libraries are available for this. The algorithm of [7] would be substantially harder to implement and it has worse constants hidden in the O-notation. The algorithm of [5] for single source shortest paths is implementable and we expect that it would work good in practice.

As mentioned before, it is not obvious that the minimum-separation problem can be solved in polynomial time. In particular, the conference version [10] of [9] gave 2-approximation algorithm for the problem. Cabello and Giannopoulos [2] provide an exact algorithm that takes $O(n^3)$ worst-case time and works for arbitrary shapes, not just disks. In this paper we improve this last algorithm to near-quadratic time for the special case of unit disks. The basic principle of the algorithm is the same, but several additional tools from Computational Geometry exploiting that we have unit disks have to be employed to reduce the worst-case running time. To show that the algorithm is *practical*, we implemented a variant of our new, near-quadratic-time algorithm and report on the experiments.

We did not aim at making a careful comparison of the available algorithms for shortest-path trees in unit disk graphs, or to develop new algorithms that could be engineered into code that runs faster. To implement our new algorithm for the minimum-separation problem we needed an implementation to compute shortest-path trees in unit disks. At the time of implementing, the algorithm [5] was not available, so we decided to implement the algorithm of [3], the most reasonable choice with optimal running time at that moment.

Assumptions. We will assume that *unit disk* means that it has radius 1/2. Up to scaling the input data, this choice is irrelevant. However, it is convenient for the exposition because then the disks intersect whenever the distance between their centres is 1. The implementation and the experiments also make this assumption.

Henceforth P will be the set of centres of $\mathcal{D}$. All the computation will be concentrated on P. In particular, we assume that P is known. (For the shortest path problem, one could possibly consider weaker models based on adjacencies.)

We will work with the graph $G_{\leqslant 1}(P)$ with vertex set P and an edge between two points $p, q \in P$ whenever their Euclidean distance $|pq|$ is at most 1. In the notation we remove the dependency on P and on the distance. Thus we just use G instead of $G_{\leqslant 1}(P)$. For simplicity of the theoretical exposition we will sometimes assume that G is connected. It is trivial to adapt to the general case, for example treating each connected component separately. The implementation does not make this assumption.

Organization of the paper. In Section 2 we discuss the theoretical algorithms for both problems and their guarantees. In Section 3 we discuss the implementations and the experimental results.

Please cite this article in press as: S. Cabello, L. Milinković, Two optimization problems for unit disks, Comput. Geom. (2018), https://doi.org/10.1016/j.comgeo.2017.12.001