Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Practical and efficient algorithms for the geometric hitting set problem

Norbert Bus [a], Nabil H. Mustafa [a,*], Saurabh Ray [b]

[a] Université Paris-Est, LIGM, Equipe A3SI, ESIEE, Paris, France
[b] Computer Science Department, New York University, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Article history:
Received 1 September 2015
Received in revised form 15 September 2017
Accepted 13 December 2017
Available online xxxx

Keywords:
Geometric hitting sets
Approximation algorithms
Computational geometry

ABSTRACT

The geometric hitting set problem is one of the basic geometric combinatorial optimization problems: given a set P of points and a set $\mathcal{D}$ of geometric objects in the plane, the goal is to compute a small-sized subset of P that hits all objects in $\mathcal{D}$. Recently Agarwal and Pan (2014) presented a near-linear time algorithm for the case where $\mathcal{D}$ consists of disks in the plane. The algorithm uses sophisticated geometric tools and data structures with large resulting constants. In this paper, we design a hitting-set algorithm for this case without the use of these data-structures, and present experimental evidence that our new algorithm has near-linear running time in practice, and computes hitting sets within 1.3-factor of the optimal hitting set. We further present `dnet`, a public source-code module that incorporates this improvement, enabling fast and efficient computation of small-sized hitting sets in practice.

1. Introduction

The minimum hitting set problem is one of the fundamental combinatorial optimization problems: given a set system $(P, \mathcal{D})$ consisting of a set P and a set $\mathcal{D}$ of subsets of P (sometimes also called *ranges*), the task is to compute the smallest subset $Q \subseteq P$ that has a non-empty intersection with each of the ranges in $\mathcal{D}$. This problem is strongly NP-hard and if there are no restrictions on the set system $\mathcal{D}$, then it is known that it is NP-hard to approximate the minimum hitting set within a logarithmic factor of the optimal [23].

The problem is NP-complete even for the case where each range has exactly two points, since this problem is equivalent to the vertex cover problem which is known to be NP-complete [16,12]. A natural case of the hitting set problem occurs when the range space $\mathcal{D}$ is derived from geometry—e.g., given a set P of n points in $\mathbb{R}^2$, and a set $\mathcal{D}$ of m triangles containing points of P, compute a minimum-sized subset of P that hits all the triangles in $\mathcal{D}$. Unfortunately, for most natural geometric set systems, computing the minimum-sized hitting set remains NP-hard. For example, even the (relatively) simple case where $\mathcal{D}$ is a set of unit disks in the plane is strongly NP-hard [15]. Note that this problem is equivalent to the problem of covering a given set of points in the plane with a minimum number of given unit disks.

Given a set system $(P, \mathcal{D})$, a positive measure μ on P (e.g., the counting measure), and a parameter $\epsilon > 0$, an ϵ-net is a subset $S \subseteq P$ such that $D \cap S \neq \emptyset$ for all $D \in \mathcal{D}$ with $\mu(D \cap P) \geq \epsilon \cdot \mu(P)$. The ϵ-net theorem [14,19] implies that for a large family of geometric set systems – balls in $\mathbb{R}^d$, half-spaces in $\mathbb{R}^d$, k-sided polytopes, r-admissible set of regions in $\mathbb{R}^d$ – there

* Corresponding author.
E-mail addresses: busn@esiee.fr (N. Bus), mustafan@esiee.fr (N.H. Mustafa), saurabh.ray@nyu.edu (S. Ray).

https://doi.org/10.1016/j.dam.2017.12.018

exist ϵ-nets of size $O\left(\frac{1}{\epsilon}\log\frac{1}{\epsilon}\right)$ [14,17]. We refer the reader to [21] for details on ϵ-nets. For certain range spaces, one can even show the existence of ϵ-nets of size $O\left(\frac{1}{\epsilon}\right)$—an important case being that of disks in $\mathbb{R}^2$ [22].

In 1994, Brönnimann and Goodrich [5] (see also [10]) proved the following interesting connection between the hitting-set problem and ϵ-nets: if one can compute an ϵ-net of size $\frac{c}{\epsilon}$ for a given set system $(P, \mathcal{D})$ in polynomial time, then one can compute a hitting set of size at most $c\cdot$Opt for $(P, \mathcal{D})$ in polynomial time, where Opt is the size of the optimal (smallest) hitting set. Until very recently, the best algorithms based on this observation had running times of $\Omega(n^2)$, and it had been a long-standing open problem to compute a $O(1)$-approximation to the hitting-set problem for disks in the plane in near-linear time. In a recent breakthrough, Agarwal and Pan [4] presented the first near-linear algorithm for computing $O(1)$-approximations for hitting sets for disks.

One limitation of this technique is that the quality of the solution is a function of the size of the ϵ-net, and so the technique cannot give better than constant-factor approximations. This limitation was overcome using an entirely different technique: local search [20,8,3]. It has been shown [20] that the local search algorithm for the hitting set problem for disks in the plane gives a PTAS. Unfortunately the running time of the algorithm to compute a $(1+\epsilon)$-approximation is $n^{O(\frac{1}{\epsilon^2})}$. Based on local search, an $\tilde{O}(n^{2.34})$ time algorithm was proposed in [7] yielding an $(8+\epsilon)$-approximation factor.

Our contributions

All approaches towards approximating geometric hitting sets for disks have to be evaluated on the questions of computational efficiency as well as approximation quality. In spite of all the progress, there remains a large gap, mainly due to the trade-offs between running times and approximation factors. The breakthrough algorithm of Agarwal and Pan [4], henceforth referred to as the AP algorithm, uses complicated data-structures that have large constants in the running time. In particular, it uses a $O(\log n+k)$-time algorithm for range reporting for disk ranges in the plane (alternatively, for halfspaces in $\mathbb{R}^3$) as well as a dynamic data-structure for maintaining approximate weighted range-counting under disk ranges in poly-logarithmic time. These data structures are based on rather sophisticated machinery, including shallow cuttings and the shallow partition theorem. We refer the reader to the survey [2] for the current state-of-the-art on range searching. These typically involve large constants in their construction and use; perhaps due to their practical shortcomings, we have not been able to find efficient (in fact, any) implementations of any of these data-structures. In practice, methods based on spatial partitioning are commonly used instead, e.g., quad-trees, *kd*-trees, R-trees and Box-trees. Many of these practical structures are suited for orthogonal searching problems; indeed, the principal library of geometric algorithms – CGAL – contains efficient algorithms for the orthogonal (called 'windowed') queries, but none for the more general and harder problem of arbitrary half-space range queries.

This work is an attempt to address this shortcoming: based on a practical spatial partitioning data structure tailored to the specific problem at hand, we give a new modified elementary algorithm and implement a variant of the algorithm that works well in practice to compute small-sized hitting sets in near-linear time, though with weaker theoretical guarantees: the worst-case running times are quadratic, while experiments indicated near-linear running times. In fact, it will turn out that an efficient practical solution for the geometric hitting set problem for disks relies on one of the basic structures in the study of planar geometry: Delaunay triangulations. A major advantage of Delaunay triangulations is that their behavior has been extensively studied, there are many efficient implementations available, and they exhibit good behavior for various real-world data-sets as well as random point sets. For computation of ϵ-nets, we will rely on the following result:

Theorem 1.1 (*[6]*). *Given a set P of n points in $\mathbb{R}^2$ and disk ranges, an ϵ-net of size at most $\frac{13.4}{\epsilon}$ can be computed in expected time $O(n\log n)$.*

Broadly, the algorithm for computing the ϵ-net in the above theorem is the following: first pick a sample Q of size $\Theta(\frac{1}{\epsilon})$. All disks that are not hit by Q are contained in two of the Delaunay disks in the Delaunay triangulation of Q. Thus, it suffices to recursively build a $\frac{\epsilon}{2}$-net for the points in each Delaunay disk. The union of all these nets together with Q is the required ϵ-net.

As an additional benefit, the algorithm used for computing ϵ-nets uses the same Delaunay triangulation as our algorithm, enabling us to reduce computations. More precisely, our contributions are:

1. A hitting set algorithm (Section 2). We present a modification of the algorithm of Agarwal and Pan that does not use any complicated data-structures—just Delaunay triangulations (and point-location on it), ϵ-nets and binary search. For example, it turns out that output sensitive range reporting is not required. This comes with a price: although experimental results indicate a near-linear running time, we have been unable to formally prove that the algorithm runs in expected near-linear time.
2. Implementation and experimental evaluation (Section 3). We present `dnet`, a public source-code module to efficiently compute small-sized hitting sets in practice. We give detailed experimental results on both synthetic and real-world data sets, which indicates that the algorithm computes, on average, a 1.3-approximation in near-linear time.

Please cite this article in press as: N. Bus, et al., Practical and efficient algorithms for the geometric hitting set problem, Discrete Applied Mathematics (2018), https://doi.org/10.1016/j.dam.2017.12.018.