Applications of Geometric Spanner Networks

Joachim Gudmundsson, National ICT Australia Ltd, www.gudmundsson.biz
Giri Narasimhan, Florida International University, www.cis.fiu.edu/~giri
Michiel Smid, Carleton University, www.scs.carleton.ca/~michiel

entry editor: Andrzej Lingas

INDEX TERMS: distance oracles, spanners, dilation, shortest paths, approximation algorithms, cluster graphs

1 PROBLEM DEFINITION

Given a geometric graph in d-dimensional space, it is useful to preprocess it so that distance queries, exact or approximate, can be answered efficiently. Algorithms that can report distance queries in constant time are also referred to as "distance oracles". With unlimited preprocessing time and space, it is clear that exact distance oracles can be easily designed. This entry sheds light on the design of approximate distance oracles with limited preprocessing time and space for the family of geometric graphs with constant dilation.

Notation and Definitions If p and q are points in $\mathbb{R}^d$, then we use the notation $|pq|$ to denote the Euclidean distance between p and q; we use the notation $\delta_G(p,q)$ to denote the Euclidean length of a shortest path between p and q in a geometric network G. Given a constant $t > 1$, a graph G with vertex set S is a t-spanner for S if $\delta_G(p,q) \leq t|pq|$ for any two points p and q of S. A t-spanner network is said to have *dilation* (or *stretch factor*) t. We also define a $(1+\varepsilon)$-approximate shortest path between p and q to be any path in G between p and q having length Δ, where $\delta_G(p,q) \leq \Delta \leq (1+\varepsilon)\delta_G(p,q)$. For a comprehensive overview of geometric spanners, see the book by Narasimhan and Smid [23].

All networks considered in this entry are simple and undirected. The model of computation used is the traditional algebraic computation tree model with the added power of indirect addressing. In particular, the algorithms presented here do not use the non-algebraic floor function as a unit-time operation. The problem is formalized below.

Problem 1 (Distance Oracle). *Given an arbitrary real constant $\epsilon > 0$, and a geometric graph G in d-dimensional Euclidean space with constant dilation t, design a data structure that answers $(1+\epsilon)$-approximate shortest path length queries in constant time.*

The data structure can also be applied to solve several other problems. These include (a) the problem of reporting approximate distance queries between vertices in a planar polygonal domain with "rounded" obstacles, (b) query versions of *closest pair* problems, and (c) the efficient computation of the approximate dilations of geometric graphs.

Survey of Related Research The design of efficient data structures for answering distance queries for general (non-geometric) networks was considered by Thorup and Zwick [26] (unweighted general graphs), Baswanna and Sen [4] (weighted general graphs, i.e., arbitrary metrics), and Arikati et al. [3] and Thorup [25] (weighted planar graphs).

For the geometric case, variants of the problem have been considered in a number of papers, including the ones by Guibas and Hershberger [17] (simple polygons), and Chen [6], Mitchell [22], Kapoor et al. [21], Hershberger and Suri [20], and Chen et al. [7] (polygonal domains with polygonal obstacles). Work on the approximate version of these variants can be found in Clarkson [9], Chen [6], Arikati et al. [3], Chen et al. [8], Har-Peled [18], Agarwal et al. [2], and Agarwal et al. [1]. The focus of this entry are the results reported in the work of Gudmundsson et al. [13–16].

2 KEY RESULTS

The main result of this entry is the existence of approximate distance oracle data structures for geometric networks with constant dilation (see Theorem 4 below). As preprocessing, the network is "pruned" so that it only has a linear number of edges. The data structure consists of a series of "cluster graphs" of increasing coarseness each of which helps answer approximate queries for pairs of points with interpoint distances of different scales. In order to pinpoint the appropriate cluster graph to search in for a given query, the data structure uses the bucketing tool described below. The idea of using cluster graphs to speed up geometric algorithms was first introduced by Das and Narasimhan [10] and later used by Gudmundsson et al. [12] to design an efficient algorithm to compute $(1+\varepsilon)$-spanners. Similar ideas were explored by Gao et al. [11] for applications in the design of mobile networks.

Pruning If the input geometric network has a superlinear number of edges, then the preprocessing step for the distance oracle data structure involves efficiently "pruning" the network so that it has only a linear number of edges. The pruning may result in a small increase of the dilation of the spanner. The following theorem was proved by Gudmundsson et al. [16].

Theorem 1. *Let $t > 1$ and $\varepsilon' > 0$ be real constants. Let S be a set of n points in $\mathbb{R}^d$, and let $G = (S, E)$ be a t-spanner for S with m edges. There exists an algorithm to compute in $O(m+n\log n)$ time, a $(1+\varepsilon')$-spanner of G having $O(n)$ edges and whose weight is $O(wt(MST(S)))$.*

The pruning step requires the following technical theorem proved by Gudmundsson et al. [16].

Theorem 2. *Let S be a set of n points in $\mathbb{R}^d$, and let $c \geq 7$ be an integer constant. In $O(n\log n)$ time, we can compute a data structure $D(S)$ consisting of:*

1. *a sequence $L_1, L_2, \ldots, L_\ell$ of real numbers, where $\ell = O(n)$, and*
2. *a sequence $S_1, S_2, \ldots, S_\ell$ of subsets of S such that $\sum_{i=1}^{\ell} |S_i| = O(n)$,*

such that the following holds. For any two distinct points p and q of S, we can compute in $O(1)$ time an index i with $1 \leq i \leq \ell$ and two points x and y in S_i such that (a) $L_i/n^{c+1} \leq |xy| < L_i$, and (b) both $|px|$ and $|qy|$ are less than $|xy|/n^{c-2}$.

Despite its technical nature, the above theorem is of fundamental importance to this work. In particular, it helps to deal with networks where the interpoint distances are not confined to a polynomial range, i.e., there are pairs of points that are very close to each other and very far from each other.

Bucketing Since the model of computation assumed here does not allow the use of floor functions, an important component of the algorithm is a "bucketing tool" that allows (after appropriate preprocessing) constant-time computation of a quantity referred to as BIndex, which is defined to be the floor of the logarithm of the interpoint distance between any pair of input points.

Theorem 3. *Let S be a set of n points in $\mathbb{R}^d$ that are contained in the hypercube $(0, n^k)^d$, for some positive integer constant k, and let ε be a positive real constant. We can preprocess S in $O(n \log n)$ time into a data structure of size $O(n)$, such that for any two points p and q of S, with $|pq| \geq 1$, we can, in constant time, compute* $\text{BINDEX}_\varepsilon(p, q) = \lfloor \log_{1+\varepsilon} |pq| \rfloor$.

The constant-time computation mentioned in Theorem 3 is achieved by reducing the problem to one of answering least common ancestor queries for pairs of nodes in a tree, a problem for which constant-time solutions were devised by Harel and Tarjan [19], Schieber and Vishkin [24], and more recently, by Bender and Farach-Colton [5].

Main Results Using the bucketing and the pruning tools, and using the algorithms described by Gudmundsson et al. [15], the following theorem can be proved.

Theorem 4. *Let $t > 1$ and $\varepsilon > 0$ be real constants. Let S be a set of n points in $\mathbb{R}^d$, and let $G = (S, E)$ be a t-spanner for S with m edges. The graph G can be preprocessed into a data structure of size $O(n \log n)$ in time $O(mn \log n)$, such that for any pair of query points $p, q \in S$, we can compute a $(1 + \varepsilon)$-approximation of the shortest-path distance in G between p and q in $O(1)$ time. Note that all the big-Oh notations hide constants that depend on d, t and ε.*

Additionally, if the traditional algebraic model of computation (without indirect addressing) is assumed, the following weaker result can be proved.

Theorem 5. *Let S be a set of n points in $\mathbb{R}^d$, and let $G = (S, E)$ be a t-spanner for S, for some real constant $t > 1$, having m edges. Assuming the algebraic model of computation, in $O(m \log \log n + n \log^2 n)$ time, we can preprocess G into a data structure of size $O(n \log n)$, such that for any two points p and q in S, we can in $O(\log \log n)$ time compute a $(1 + \varepsilon)$-approximation of the shortest-path distance in G between p and q.*

3 APPLICATIONS

As mentioned earlier, the data structure described above can be applied to several other problems. The first application deals with reporting distance queries for a planar domain with polygonal obstacles. The domain is further constrained to be t-rounded, which means that the length of the shortest obstacle-avoiding path between any two points in the input point set is at most t times the Euclidean distance between them. In other words, the visibility graph is required to be a t-spanner for the input point set.

Theorem 6. *Let $\mathcal{F}$ be a t-rounded collection of polygonal obstacles in the plane of total complexity n, where t is a positive constant. One can preprocess $\mathcal{F}$ in $O(n \log n)$ time into a data structure of size $O(n \log n)$ that can answer obstacle-avoiding $(1 + \varepsilon)$-approximate shortest path length queries in time $O(\log n)$. If the query points are vertices of $\mathcal{F}$, then the queries can be answered in $O(1)$ time.*

The next application of the distance oracle data structure includes query versions of *closest pair* problems, where the queries are confined to specified subset(s) of the input set.

Theorem 7. *Let $G = (S, E)$ be a geometric graph on n points and m edges, such that G is a t-spanner for S, for some constant $t > 1$. One can preprocess G in time $O(m + n \log n)$ into a data structure of size $O(n \log n)$ such that given a query subset S' of S, a $(1 + \varepsilon)$-approximate closest pair in S' (where distances are measured in G) can be computed in time $O(|S'| \log |S'|)$.*

Theorem 8. *Let $G = (S, E)$ be a geometric graph on n points and m edges, such that G is a t-spanner for S, for some constant $t > 1$. One can preprocess G in time $O(m + n \log n)$ into a data structure of size $O(n \log n)$ such that given two disjoint query subsets X and Y of S, a $(1 + \varepsilon)$-approximate bichromatic closest pair (where distances are measured in G) can be computed in time $O((|X| + |Y|) \log(|X| + |Y|))$.*

The last application of the distance oracle data structure includes the efficient computation of the approximate dilations of geometric graphs.

Theorem 9. *Given a geometric graph on n vertices with m edges, and given a constant C that is an upper bound on the dilation t of G, it is possible to compute a $(1+\varepsilon)$-approximation to t in time $O(m + n \log n)$.*

4 OPEN PROBLEMS

Two open problems remain unanswered.

1. Improve the space utilization of the distance oracle data structure from $O(n \log n)$ to $O(n)$.
2. Extend the approximate distance oracle data structure to report not only the approximate distance, but also the approximate shortest path between the given query points.

5 CROSS REFERENCES

LINK TO ENTRIES BY GUDMUNDSSON, NARASIMHAN AND SMID; LINK TO ENTRIES ON SHORTEST PATH, IF ANY.

References

[1] P. K. Agarwal, S. Har-Peled, and M. Karia, *Computing approximate shortest paths on convex polytopes*, in Proceedings of the 16th ACM Symposium on Computational Geometry, 2000, pp. 270–279.

[2] P. K. Agarwal, S. Har-Peled, M. Sharir, and K. R. Varadarajan, *Approximate shortest paths on a convex polytope in three dimensions*, Journal of the ACM, 44 (1997), pp. 567–584.

[3] S. Arikati, D. Z. Chen, L. P. Chew, G. Das, M. Smid, and C. D. Zaroliagis, *Planar spanners and approximate shortest path queries among obstacles in the plane*, in Proceedings of the 4th Annual European Symposium on Algorithms, vol. 1136 of Lecture Notes in Computer Science, Berlin, 1996, Springer-Verlag, pp. 514–528.

[4] S. Baswana and S. Sen, *Approximate distance oracles for unweighted graphs in $\tilde{O}(n^2)$ time*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 271–280.

[5] M. A. Bender and M. Farach-Colton, *The LCA problem revisited*, in Proceedings of the 4th Latin American Symposium on Theoretical Informatics, vol. 1776 of Lecture Notes in Computer Science, Berlin, 2000, Springer-Verlag, pp. 88–94.

[6] D. Z. Chen, *On the all-pairs Euclidean short path problem*, in Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms, 1995, pp. 292–301.

[7] D. Z. Chen, O. Daescu, and K. S. Klenk, *On geometric path query problems*, International Journal of Computational Geometry & Applications, 11 (2001), pp. 617–645.

[8] D. Z. Chen, K. S. Klenk, and H.-Y. T. Tu, *Shortest path queries among weighted obstacles in the rectilinear plane*, SIAM Journal on Computing, 29 (2000), pp. 1223–1246.

[9] K. L. Clarkson, *Approximation algorithms for shortest path motion planning*, in Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 56–65.

[10] G. Das and G. Narasimhan, *A fast algorithm for constructing sparse Euclidean spanners*, International Journal of Computational Geometry & Applications, 7 (1997), pp. 297–315.

[11] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, *Discrete mobile centers*, Discrete & Computational Geometry, 30 (2003), pp. 45–63.

[12] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan, *Fast greedy algorithms for constructing sparse geometric spanners*, SIAM Journal on Computing, 31 (2002), pp. 1479–1500.

[13] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. Smid, *Approximate distance oracles for geometric graphs*, in Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms, 2002, pp. 828–837.

[14] ——, *Approximate distance oracles revisited*, in Proceedings of the 13th International Symposium on Algorithms and Computation, vol. 2518 of Lecture Notes in Computer Science, Berlin, 2002, Springer-Verlag, pp. 357–368.

[15] ——, *Approximate distance oracles for geometric spanners*, ACM Transactions on Algorithms, (2007). To Appear.

[16] J. Gudmundsson, G. Narasimhan, and M. Smid, *Fast pruning of geometric spanners*, in Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science, vol. 3404 of Lecture Notes in Computer Science, Berlin, 2005, Springer-Verlag, pp. 508–520.

[17] L. J. Guibas and J. Hershberger, *Optimal shortest path queries in a simple polygon*, Journal of Computer and System Sciences, 39 (1989), pp. 126–152.

[18] S. Har-Peled, *Approximate shortest paths and geodesic diameters on convex polytopes in three dimensions*, in Proceedings of the 13th ACM Symposium on Computational Geometry, 1997, pp. 359–365.

[19] D. Harel and R. E. Tarjan, *Fast algorithms for finding nearest common ancestors*, SIAM Journal on Computing, 13 (1984), pp. 338–355.

[20] J. Hershberger and S. Suri, *An optimal algorithm for Euclidean shortest paths in the plane*, SIAM Journal on Computing, 28 (1999), pp. 2215–2256.

[21] S. Kapoor, S. N. Maheshwari, and J. S. B. Mitchell, *An efficient algorithm for Euclidean shortest paths among polygonal obstacles in the plane*, Discrete & Computational Geometry, 18 (1997), pp. 377–383.

[22] J. S. B. Mitchell, *Shortest paths among obstacles in the plane*, International Journal of Computational Geometry & Applications, 6 (1996), pp. 309–332.

[23] G. Narasimhan and M. Smid, *Geometric Spanner Networks*, Cambridge University Press, Cambridge, UK, 2007.

[24] B. Schieber and U. Vishkin, *On finding lowest common ancestors: simplifications and parallelisations*, SIAM Journal on Computing, 17 (1988), pp. 327–334.

[25] M. Thorup, *Compact oracles for reachability and approximate distances in planar digraphs*, Journal of the ACM, 51 (2004), pp. 993–1024.

[26] M. Thorup and U. Zwick, *Approximate distance oracles*, in Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing, 2001, pp. 183–192.