

Computing the Maximum Detour and Spanning Ratio of Planar Chains, Trees and Cycles*

Stefan Langerman[†]

Pat Morin[†]

Michael Soss[†]

1 Introduction

Let $G = (V, E)$ be an embedded connected graph with n vertices and m edges. Specifically, the vertex set V consists of points in \mathbb{R}^2 , and E consists of closed line segments whose endpoints are in V . Let $SP(G, s, t)$ denote the length of the shortest path from s to t in $\cup E$. The detour between two points $s, t \in \cup E$ is defined as

$$D(G, s, t) = \frac{SP(G, s, t)}{|st|},$$

where $|st|$ denotes the Euclidean distance between points s and t . The *maximum detour* $D(G)$ of G is the maximum detour over all pairs of points in G , i.e.,

$$D(G) = \max\{D(G, s, t) : s, t \in \cup E, s \neq t\}.$$

The *spanning ratio* or *stretch factor* $S(G)$ of G is the maximum detour over all pairs of vertices of G , i.e.,

$$S(G) = \max\{D(G, s, t) : s, t \in V, s \neq t\}.$$

The maximum detour and spanning ratio play important roles in the analysis of online routing algorithms [1, 6] and the construction of spanners [5]. In the former case, the goal is to find paths that minimize maximum detour. In the latter, the goal is to construct graphs with few edges that minimize the spanning ratio.

2 Previous Work

Recently, researchers have become interested in computing the maximum detour and spanning ratio of embedded graphs. The spanning ratio can be computed in $O(n(m + n \log n))$ time by computing the shortest paths between all pairs of vertices and then comparing these to the distances between all pairs of vertices. In \mathbb{R}^2 , the maximum detour is infinite if G is non planar, so the maximum detour can be computed in $O(n^2 \log n)$ time by computing shortest paths and using this information to find the maximum detour between each pair of edges. Surprisingly, these are the best known results for computing the maximum detour or spanning ratio. Even if the input graph G is a polygonal chain, no sub-quadratic time algorithms are known, though fast approximation algorithms have been reported.

Narasimhan and Smid [8] give $O(n \log n)$ time algorithms that can ϵ -approximate the spanning ratio of G when G is a path, a cycle or a tree. More generally, they show that, after $O(n \log n)$ preprocessing, the problem of approximating the spanning ratio can be reduced to $O(n)$ approximate shortest path queries on G . Their results make use of the well-separated pair decomposition of Callahan and Kosaraju [2] and hold even for graphs embedded in \mathbb{R}^d . The authors also show that approximating the spanning ratio requires $\Omega(n \log n)$ time in the algebraic decision tree model of computation, even for chains in \mathbb{R}^1 .

Ebbers-Baumann *et al* [4] study the problem of approximating the maximum detour of a polygonal chain, and give an $O(n \log n)$ time algorithm that finds an ϵ -approximation to the maximum detour. Their result is based on the θ -graph spanner of Keil

*This research was partly funded by NSERC.

[†]School of Computer Science, McGill University, 3480 University St., Suite 318, Montréal, Québec, CANADA, H3A 2A7. {morin, godfried}@cgm.cs.mcgill.ca

and Gutwin [7].

3 New Results

In this paper we give randomized algorithms with $O(n \log n)$ expected running time that compute the exact spanning ratio or maximum detour of a polygonal chain with n vertices. These are the first sub-quadratic time algorithms for finding the exact spanning ratio or maximum detour, and they solve open problems posed in at least two papers [4, 8].

We solve these problems by reducing the associated decision problem to computing the upper envelope of a set of identical cones in \mathbb{R}^3 . In the case of spanning ratio, the set of cones is finite, and the upper envelope that we compute is actually an additively-weighted Voronoi diagram of points in the plane. In the case of maximum detour, the set of cones is infinite, and corresponds to computing the additively-weighted Voronoi diagram of line segments in the plane, a diagram that seems not to have been considered previously. We then apply a general optimization technique of Chan [3] to convert the decision algorithm into an optimization algorithm.

We also show that more complicated structures can sometimes be treated by using multiple invocations of the above technique. As examples, we give an $O(n \log^2 n)$ time algorithm for computing the maximum detour or spanning ratio of a planar tree, an $O(\frac{n}{\epsilon} \log n)$ time algorithm that ϵ -approximates the maximum detour on a planar cycle, and an $O(n^{\frac{3}{2}} \log n)$ time algorithm for computing the maximum detour or spanning ratio of a planar cycle.

The full version of the paper, which includes all details, can be found online at the second author's web page <http://cgm.cs.mcgill.ca/~morin/publications>.

References

[1] P. Bose and P. Morin. Competitive online routing in geometric graphs. In *Proceedings of the VIII International Colloquium on Structural Informa-*

tion and Communication Complexity (SIROCCO 2001), 2001.

- [2] P. B. Callahan and S. R. Kosaraju. A decomposition of multi-dimension point-sets with applications to k -nearest neighbours and N -body potential fields. *Journal of the ACM*, 42:67–90, 1995.
- [3] T. M. Chan. Geometric applications of a randomized optimization technique. *Discrete & Computational Geometry*, 22(4):547–567, 1999.
- [4] A. Ebberts-Baumann, R. Klein, E. Langetepe, and A. Lingas. A fast algorithm for approximating the detour of a polygonal chain. In *Proceedings of the 9th Annual European Symposium on Algorithms (ESA 2001)*, pages 321–332, 2001.
- [5] D. Eppstein. Spanning trees and spanners. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. Elsevier, 1999.
- [6] C. Icking and R. Klein. Searching for the kernel of a polygon: A competitive strategy. In *Proceedings of the 11th Annual Symposium on Computational Geometry*, pages 258–266, 1995.
- [7] J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discrete & Computational Geometry*, 7:13–28, 1992.
- [8] G. Narasimhan and M. Smid. Approximating the stretch factor of Euclidean graphs. *SIAM Journal on Computing*, 30(3):978–989, 2001.