WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

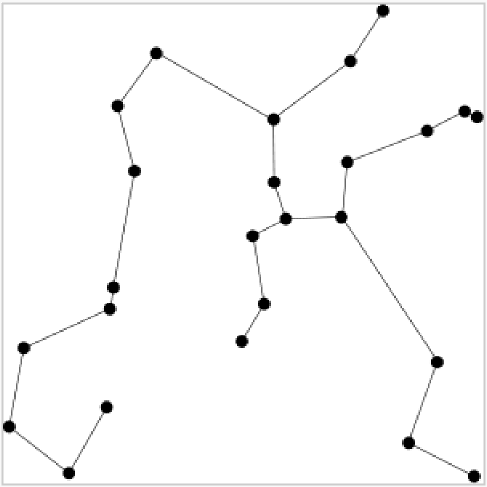
Print/export
Create a book
Download as PDF
Printable version

Languages
فارسی
Српски / srpski
ไทย
Edit links

Euclidean minimum spanning tree

From Wikipedia, the free encyclopedia

The **Euclidean minimum spanning tree** or **EMST** is a minimum spanning tree of a set of *n* points in the plane (or more generally in I^{d}), where the weight of the edge between each pair of points is the Euclidean distance between those two points. In simpler terms, an EMST connects a set of dots using lines such that the total length of all the lines is minimized and any dot can be reached from any other by following the lines.



An EMST of 25 random points in the plane

In the plane, an EMST for a given set of points may be found in Θ(*n* log *n*) time using O(*n*) space in the algebraic decision tree model of computation. Faster randomized algorithms of complexity O(*n* log log *n*) are known in more powerful models of computation that more accurately model the abilities of real computers.[1]

In higher dimensions (*d* ≥ 3), finding an optimal algorithm remains an open problem.

Contents [hide]

Lower bound [edit]

An asymptotic lower bound of Ω(*n* log *n*) for time complexity of the EMST problem can be established in restricted models of computation, such as the algebraic decision tree and algebraic computation tree models, in which the algorithm has access to the input points only through certain restricted primitives that perform simple algebraic computations on their coordinates: in these models, the closest pair of points problem requires Ω(*n* log *n*) time, but the closest pair is necessarily an edge of the EMST, so the EMST also requires this much time.[2] However, if the input points have integer coordinates and bitwise operations and table indexing operations are permitted using those coordinates, then faster algorithms are possible.[1]

Algorithms for computing EMSTs in two dimensions [edit]

The simplest algorithm to find an EMST in two dimensions, given *n* points, is to actually construct the complete graph on *n* vertices, which has *n*(*n*-1)/2 edges, compute each edge weight by finding the distance between each pair of points, and then run a standard minimum spanning tree algorithm (such as the version of Prim's algorithm or Kruskal's algorithm) on it. Since this graph has Θ(n^2) edges for *n* distinct points, constructing it already requires Ω(n^2) time. This solution also requires Ω(n^2) space to store all the edges.

A better approach to finding the EMST in a plane is to note that it is a subgraph of every Delaunay triangulation of the *n* points, a much-reduced set of edges:

1. Compute the Delaunay triangulation in O(*n* log *n*) time and O(*n*) space. Because the Delaunay triangulation is a planar graph, and there are no more than three times as many edges as vertices in any planar graph, this generates only O(*n*) edges.
2. Label each edge with its length.

3. Run a graph minimum spanning tree algorithm on this graph to find a minimum spanning tree. Since there are O(*n*) edges, this requires O(*n* log *n*) time using any of the standard minimum spanning tree algorithms such as Borůvka's algorithm, Prim's algorithm, or Kruskal's algorithm.

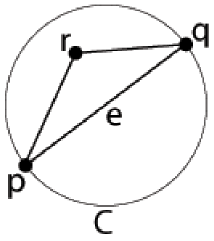
The final result is an algorithm taking O(*n* log *n*) time and O(*n*) space.

If the input coordinates are integers and can be used as array indices, faster algorithms are possible: the Delaunay triangulation can be constructed by a randomized algorithm in O(*n* log log *n*) expected time.[1] Additionally, since the Delaunay triangulation is a planar graph, its minimum spanning tree can be found in linear time by a variant of Borůvka's algorithm that removes all but the cheapest edge between each pair of components after each stage of the algorithm.[3] Therefore, the total expected time for this algorithm is O(*n* log log *n*).[1]

Higher dimensions [edit]

The problem can also be generalized to *n* points in the *d*-dimensional space I^{d}. In higher dimensions, the connectivity determined by the Delaunay triangulation (which, likewise, partitions the convex hull into *d*-dimensional simplices) contains the minimum spanning tree; however, the triangulation might contain the complete graph.[4] Therefore, finding the Euclidean minimum spanning tree as a spanning tree of the complete graph or as a spanning tree of the Delaunay triangulation both take O(dn^2) time. For three dimensions it is possible to find the minimum spanning tree in time O((n log n)$^{4/3}$), and in any dimension greater than three it is possible to solve it in a time that is faster than the quadratic time bound for the complete graph and Delaunay triangulation algorithms.[4] For uniformly random point sets it is possible to compute minimum spanning trees as quickly as sorting.[5] Using a well-separated pair decomposition, it is possible to produce a (1+ε)-approximation in O(*n log n*) time.[6]

Subtree of Delaunay triangulation [edit]



All edges of an EMST are edges of a relative neighborhood graph,[7][8][9] which in turn are edges of a Gabriel graph, which are edges in a Delaunay triangulation of the points,[10][11] as can be proven via the equivalent contrapositive statement: *every edge not in a Delaunay triangulation is also not in any EMST.* The proof is based on two properties of minimum spanning trees and Delaunay triangulations:

1. (the cycle property of minimum spanning trees): *For any cycle C in the graph, if the weight of an edge e of C is larger than the weights of other edges of C, then this edge cannot belong to a MST*.
2. (a property of Delaunay triangulations): *If there is a circle with two of the input points on its boundary which contains no other input points, the line between those two points is an edge of every Delaunay triangulation.*

Consider an edge *e* between two input points *p* and *q* which is not an edge of a Delaunay triangulation. Property 2 implies that the circle *C* with *e* as its diameter must contain some other point *r* inside. But then *r* is closer to both *p* and *q* than they are to each other, and so the edge from *p* to *q* is the longest edge in the cycle of points *p* → *q* → *r* → *p*, and by property 1 *e* is not in any EMST.

Expected size [edit]

The expected size of the EMST for large numbers of points has been determined by J. Michael Steele. If f is the density of the probability function for picking points, then for large n and $d \neq 1$ the size of the EMST is approximately

$$c(d)n^{\frac{d-1}{d}} \int_{\mathbb{R}^d} f(x)^{\frac{d-1}{d}} dx$$

where $c(d)$ is a constant depending only on the dimension d. The exact value of the constants are unknown but can be estimated from empirical evidence.

Applications [edit]

An obvious application of Euclidean minimum spanning trees is to find the cheapest network of wires or pipes to connect a set of places, assuming the links cost a fixed amount per unit length. However, while these give an absolute lower bound on the amount of connection needed, most such networks prefer a *k*-connected graph to a tree, so that failure of an any individual link will not split the network into parts.

Another application of EMSTs is a constant-factor approximation algorithm for approximately solving the Euclidean traveling salesman problem, the version of the traveling salesman problem on a set of points in the plane with edges labelled by their length. This realistic variation of the problem can be solved within a factor of 2 by computing the EMST, doing a walk along its boundary which outlines the entire tree, and then removing all but one occurrence of each vertex from this walk.

Planar realization [edit]

The *realization problem* for Euclidean minimum spanning trees is stated as follows: Given a tree $T = (V, E)$, find a location $D(u)$ for each vertex $u \in V$ so that T is a minimum spanning tree of $D(u)$: u ∈ V, or determine that no such locations exist. Testing of the existence of a realization in the plane is NP-hard.[12]

See also [edit]

- Rectilinear minimum spanning tree

References [edit]

1. ^ a b c d Buchin, Kevin; Mulzer, Wolfgang (2009). *Delaunay triangulations in O(sort(n)) time and more* (PDF). Proc. 50th IEEE Symposium on Foundations of Computer Science. pp. 139–148. doi:10.1109/FOCS.2009.53..
2. ^ Yao, A. C.-C. (1989), "Lower bounds for algebraic computation trees with integer inputs", *Proc. 30th Annual Symposium on Foundations of Computer Science (FOCS 1989)*, pp. 308–313, doi:10.1109/SFCS.1989.63495.
3. ^ Eppstein, David (1999), "Spanning trees and spanners", in Sack, J.-R.; Urrutia, J., *Handbook of Computational Geometry*, Elsevier, pp. 425–461; Mareš, Martin (2004), "Two linear time algorithms for MST on minor closed graph classes" (PDF), *Archivum mathematicum* **40** (3): 315–320.
4. ^ a b Agarwal, P. K.; Edelsbrunner, H.; Schwarzkopf, O.; Welzl, E. (1991), "Euclidean minimum spanning trees and bichromatic closest pairs", *Discrete and Computational Geometry* (Springer) **6** (1): 407–422, doi:10.1007/BF02574698.
5. ^ Chatterjee, S.; Connor, M.; Kumar, P. (2010), "Geometric minimum spanning trees with GeoFilterKruskal", in Festa, Paola, *Symposium on Experimental Algorithms*, Lecture Notes in Computer Science **6049**, Springer-Verlag, pp. 486–500, doi:10.1007/978-3-642-13193-6_41.
6. ^ Smid, Michiel (16 August 2005). "The well-separated pair decomposition and its applications" (PDF). Retrieved 26 March 2014.
7. ^ Jerzy W. Jaromczyk and Godfried T. Toussaint, "Relative neighborhood graphs and their relatives," *Proceedings of the IEEE*, Vol. 80, No. 9, September 1992, pp. 1502–1517.
8. ^ Godfried T. Toussaint, "Comment on algorithms for computing relative neighborhood graph," *Electronics Letters*, Vol. 16, No. 22, October 1981, pp. 860–861.
9. ^ Godfried T. Toussaint, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, Vol. 12, 1980, pp. 261–268.
10. ^ Robert Pless. Lecture 17: Voronoi Diagrams and Delauney Triangulations. Spring 2003, Computational Geometry Class Page. Associate Professor of Computer Science and Engineering at Washington University. http://www.cs.wustl.edu/~pless/506/l17.html
11. ^ Robert Sedgewick and Kevin Wayne. Minimum Spanning Tree lecture notes. Computer Science 226: Algorithms & Data Structures, Spring 2007. Princeton University. http://www.cs.princeton.edu/courses/archive/spr07/cos226/lectures/19MST.pdf
12. ^ Eades, Peter; Whitesides, Sue (1994), "The realization problem for Euclidean minimum spanning trees is NP-hard", *Proc. 10th ACM Symposium on Computational Geometry*, pp. 49–56, doi:10.1145/177424.177507.

- Smith College: The Open Problems Project: Problem 5: Euclidean Minimum Spanning Tree
- Max-Planck-Institut fuer Informatik: Exercise solutions, by Kavitha Telikepalli (Postscript)
- STANN (Michael Connor, Piyush Kumar and Samidh Chatterjee): A C++ library that can compute Euclidean Minimum Spanning Trees in low dimensions

Categories: Spanning tree | Geometric graphs