An Exploratory Assignment on Minimum Spanning Trees

Arnau Abella
Universitat Politècnica de Catalunya

January 1, 2021

1 Introduction

Let the weight of a tree be the sum of the squares of its edges lengths. Given a set of points P in the unit square $I \times I$, let $W(P)$ be the weight of the *minimum spanning tree (MST)* of P, where an edge length is the *Euclidean distance* between its endpoints. If P consist of the four corners of the square, then $W(P) = 3$. Gilbert and Pollack [1] proved that $W(P)$ is $\mathcal{O}(1)$ and this was extended to an arbitrary number of dimensions by Bern and Eppstein [2]. While more recent *divide-and-conquer* approaches have show that $W(P) \leq 4$, no point set is known with $W(P) > 3$, and hence it has been widely conjectured that $W(P) \leq 3$. In 2013, it was proven that $W(P) < 3.41$ [3]. Here we show an empirical experiment to check whether $W(P) < 3.41$ holds for any $MST(P)$.

2 Experiment

In order to check the previous theorem, we generate uniformaly at random points in the unite square P and compute the weight of the MST. We do this with an increasing number of points in order to explore the solution space. It is important to note that the exploration is not exhaustive since exploring the whole solution space would require a large amount of computational power and the search does not explicitly aims for the degenerate instances where $W(P) > 3.41$ may happen.

These random points are generated using a pseudo-random number generator (PRNG) that uses *Marsaglia's MWC256* (also known as *MWC8222*) which has a period of 2^{8222} and fares well in tests of randomness. It is also extremely fast, between $2-3$ times faster than the *Mersenne Twister*.

Once the random points are generated, we build a complete undirected graph $G = (V, E)$ where $|V| = n$ and $|E| = \binom{n}{2}$ using an *inductive graph* representation which is efficiently implemented as a *big-endian patricia trees*.

Then, we search the *minimum spanning tree* on the inducive graph using *Prim's algorihm* (see Algorithm 1). The inductive implementation has $\mathcal{O}(|V|^2)$ time complexity which could be improved up to $\mathcal{O}(|E| + |V| \log |V|)$ using a *fibonacci heap*.

Algorithm 1: Minimum Spanning Tree Prim's Algorithm

Input: An undirected weighted graph $G = (V, E)$
Output: The minimum spanning tree of the input graph G
foreach $v \in V$ **do**
  $key(v) = \infty$
  $parent(v) = NIL$
end
$key(r) = 0$ `// Pick` ***u.a.r.*** `the initial vertex` $r \in V$
$Q = V$
while $Q \neq \emptyset$ **do**
  $u = EXTRACT - MIN(Q)$
  foreach $v \in ADJ(u)$ **do**
    if $v \in Q$ *and* $w(u, v) < key(v)$ **then**
      $key(v) = w(u, v)$
      $parent(v) = u$
    end
  end
end

We emphasise that there are more efficient algorithms such as Karger, Klein and Tarjan linear-time randomized algorithm [4] or Bernand Chazelle almost linear algorithm [5] but we decided to use Prim's algorithm because it is the simplest to implement and the performance was good enough for this experiment (see Table 1).

$\lvert V \rvert$	Time	R^2	μ	σ
128	8.1 ms	0.996	8.358 ms	327.8 μs
512	406.0 ms	1.0	391.9 ms	10.33 ms
1024	153.6 ms	1.0	150.3 ms	3.178 ms
2048	943.1 ms	1.0	934.4 ms	14.24 ms

Table 1: Prim's Algorithm benchmark.

One optimization applied to the search is the removal of edges that are unlikely to be used. The minimum spanning tree is unlikely to use any edge of weight greater than $k(n)$ for some function $k(n)$. We estimated the function $k(n)$ by calculating the MST on random graphs of increasing size in the unite square (see Table 2).

$\|V\|$	W_{max}
16	0.220
32	0.150
64	0.060
128	0.040
256	0.020
512	0.013
1024	0.008
2048	0.004

Table 2: Estimation of $k(n)$.

The code of the experiment is publicly available at `https://github.com/monadplus/mst-experiment`.

3 Results

The experiment was run on a ThinkPad T495 (AMD Ryzen 7 Pro 3700U, 13934 MiB of RAM, Linux 5.8.10-arch1-1). The experiment was implemented purely in haskell on the Glasgow Haskell Compiler (GHC) 8.8.4. The experiment was run up to 8196 vertices where the execution time increased up to $30'$ of CPU time and we could not generate reliable results.

$\|V\|$	μ_W	σ_W
16	1.1067605	0.21767415
32	1.1015033	0.20797518
64	1.2079728	0.12993012
128	1.1780577	0.10956229
256	1.2000419	0.08088318
512	1.1488761	0.02073750
1024	1.2366604	0.05549732
2048	1.2135003	0.00306511
4096	1.1810393	0.01102620

Table 3: Experiment results.

The results show that $W(P)$ is $\mathcal{O}(1)$ (see Table 3 and Figure 1). Note, the deviation decreases when the number of vertices increases. As a conclusion, we believe that there is no evidence to refute the claim $W(P) < 3.41$.

For future work, the search could be tunned to better explore the degenerate instances which would lead to a stronger position to accept or refute the theorem.

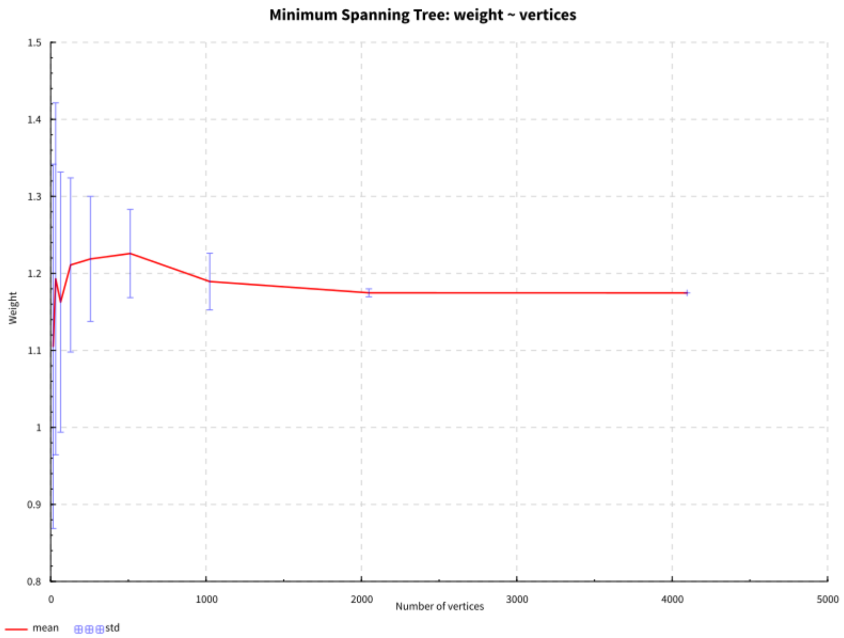

Figure 1: Experiment results.

References

[1] E. N. Gilbert and H. O. Pollack. Steiner minimal trees. *SIAM J. App. Math.*, 16(1):1–29, 1968.

[2] M. W. Bern and D. Eppstein. Worst-case bounds for subadditive geometric graphs. *In Proc. 9th Symp. on Comp. Geom.*, pages 183–188, 1993.

[3] Oswin Aichholzer, S Allen, Greg Aloupis, Luis Barba, Prosenjit Bose, JL de Varufel, John Iacono, Stefan Langerman, D Souvaine, Perouz Taslakian, et al. Sum of squared edges for mst of a point set in a unit square. In *Japanese Conference on Discrete and Computational Geometry (JCDCG)*, 2013.

[4] David R Karger, Philip N Klein, and Robert E Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM (JACM)*, 42(2):321–328, 1995.

[5] Bernard Chazelle. A minimum spanning tree algorithm with inverse-ackermann type complexity. *Journal of the ACM (JACM)*, 47(6):1028–1047, 2000.