Low-Crossing Spanning Trees: an Alternative Proof and Experiments

Panos Giannopoulos* Maximilian Konzack† Wolfgang Mulzer*

Abstract

We give a quick proof that any planar n-point set has a spanning tree with crossing number $O(\sqrt{n})$. Our proof relies on an LP-based approach by Har-Peled [8], and it uses Farkas' lemma. We also present a new heuristic for computing a spanning tree with low crossing number and compare it experimentally with other known approaches.

1 Introduction

Let P be a planar n-point set in general position. A *spanning tree* T for P is a plane geometric graph with vertex set P that is connected and acyclic. The *crossing number* of T is the maximum number of line segments in T that can be intersected by any line.

Chazelle and Welzl showed that P always has a spanning tree with crossing number $O(\sqrt{n})$ [5]. The original proof uses *iterative reweighting* [5, 11], a method with widespread use throughout theoretical computer science that is closely connected to linear programming [3]. Har-Peled [8] made this connection explicit for spanning trees with low crossing number: their existence can be proved by showing that a certain LP is feasible and applying iterative randomized rounding (similar techniques also appear in [4, 6]). We give an alternative proof of feasibility that uses Farkas' Lemma.

The problem of computing a spanning tree with minimum crossing number is NP-hard [7], while there is an $O(\log n/\log\log n)$- and $O(\log n)$-approximation algorithm (resp.) [8, 6] and a heuristic based on iterative LP-rounding [7], see also Section 3. We present a new heuristic and perform extensive experiments.

Preliminaries and Notation. Let E_P be the set of line segments pq with $p \neq q \in P$, and L_P a set of representative lines for all possible ways of how P can be partitioned into two sets by a line; $|L_P| = O(n^2)$. For $\ell \in L_P$, let $E_\ell \subseteq E_P$ be the set of all edges that intersect ℓ. Conversely, for $pq \in E_P$, let $L_{pq} \subseteq L_P$ be the set of lines that intersect pq. Clearly, it suffices to bound the crossing number w.r.t. the lines in L_P.

Let L be a set of lines in the plane and $p, q \in \mathbb{R}^2$ two points. The *crossing distance* $d_L(p,q)$ between p and q with respect to L is the number of lines in L intersected by the line segment pq. The *crossing disk* around p with radius $r > 0$, $D_L(p,r)$, is the set of all points $q \in \mathbb{R}^2$ with $d_L(p,q) \leq r$. We will need the following lemma [11, Lemma 2.1]:

Lemma 1 *For any $p \in \mathbb{R}^2$ and $r \in \{0, \ldots, \lceil |L|/2 \rceil\}$, the disk $D_L(p,r)$ contains at least $\binom{r+1}{2}$ vertices of the arrangement of L.*

We will use the following variant of Farkas' lemma [10, Exercise 1.3.7(b)].

Lemma 2 (Farkas' Lemma) *Let A be a rational $m \times n$ matrix and $b \in \mathbb{Q}^m$. Either there is a vector $x \in \mathbb{Q}^n$ that satisfies $Ax \leq b$, $x \geq 0$, or there is a vector $y \in \mathbb{Q}^m$ that satisfies $A^T y \geq 0$, $b^T y < 0, y \geq 0$.*

2 Existence of Trees with Low Crossing Number

The following LP models a graph on P with crossing number $O(\sqrt{n})$ where each point has an incident edge.

$$\sum_{pq \in E_\ell} x_{pq} \leq \sqrt{n}, \text{ for all } \ell \in L_P$$
$$\sum_{pq \in E_P} x_{pq} \geq 1, \text{ for all } p \in P$$
$$x_{pq} \geq 0, \text{ for all } pq \in E_P.$$

Lemma 3 *The LP is feasible.*

Proof. By Lemma 2, it is enough to show that the following system is infeasible.

$$\sqrt{n} \sum_{\ell \in L_P} y_\ell < \sum_{p \in P} y_p \qquad (*)$$
$$\sum_{\ell \in L_{pq}} y_\ell \geq y_p + y_q, \text{ for all } pq \in E_P$$
$$y_\ell \geq 0, y_p \geq 0, \text{ for all } \ell \in L_P, p \in P.$$

Suppose there is a solution $y \in \mathbb{Q}^m$. From (*) we derive that there is a $c > 0$ so that for all $\lambda \in \mathbb{N}$,

$$\sqrt{n} \sum_{\ell \in L_P} \lambda y_\ell = \sum_{p \in P} \lambda y_p - \lambda c \qquad (1)$$

Fix $\lambda \in \mathbb{N}$ and set $z = \lambda y$. For λ large enough, z is integral. Let L be the line set with z_ℓ copies of ℓ, for each $\ell \in L_P$, slightly perturbed so that L contains $N = \sum_{\ell \in L_P} z_\ell$ lines in general position.

*Institut für Informatik, Freie Universität Berlin, Germany, {panos, mulzer}@inf.fu-berlin.de

†TU Eindhoven, The Netherlands, m.p.konzack@tue.nl.
Supported by NWO, project no. 612.001.207

Let $pq \in E_P$ with $z_p, z_q \geq 1$. Since z is also a solution, we have $d_L(p,q) = \sum_{\ell \in L_{pq}} z_\ell \geq z_p + z_q$. Then $D_L(p, z_p - 1) \cap D_L(q, z_q - 1) = \emptyset$: otherwise there would be a point r with $d_L(p,r) \leq z_p - 1$ and $d_L(r,q) \leq z_q - 1$, and the triangle inequality would give $d_L(p,q) \leq z_p + z_q - 2$. By Lemma 1, this implies $\sum_{p\in P} \binom{z_p}{2} \leq \binom{N}{2}$, since the arrangement of L has $\binom{N}{2}$ vertices. Hence,

$$\sum_{p\in P} z_p \leq \sqrt{n}\sqrt{\sum_{p\in P} z_p^2} \leq \sqrt{n}\sqrt{\sum_{p\in P} 2\binom{z_p}{2} + z_p}$$
$$\leq \sqrt{n}\sqrt{2\binom{N}{2} + \sum_{p\in P} z_p} \leq \sqrt{n}N + \sqrt{n\sum_{p\in P} z_p},$$

where we used Cauchy-Schwarz; $a^2 = 2\binom{a}{2} + a$; the above observation; and finally $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$. Recalling (1), $N = \sum_{\ell\in L_P} z_\ell$, and $z = \lambda y$, we get

$$\sum_{p\in P} \lambda y_p - \lambda c = \sqrt{n}\sum_{\ell\in L_P} \lambda y_\ell \geq \sum_{p\in P} \lambda y_p - \sqrt{n\sum_{p\in P} \lambda y_p}.$$

Thus, $\lambda \leq (n/c^2)\sum_{p\in P} y_p$, a contradiction to $\lambda \in \mathbb{N}$ being arbitrary. Hence, the LP must be feasible. ☐

Iterative rounding now shows that a spanning tree of low crossing number exists [8, 4, 6]. For completeness, we include a proof.

Lemma 4 *Let $P \subseteq \mathbb{R}^2$ be an n-point set in general position. There is a set $E \subseteq E_P$ of line segments so that the graph (P,E) has at most $3n/4$ components and crossing number at most $c\sqrt{n}$, for a fixed $c > 0$.*

Proof. We use the probabilistic method. Take a feasible solution x for the LP, as in Lemma 3. We include each $pq \in E_P$ in E independently with probability $\min\{x_{pq}, 1\}$. For each $p \in P$, the probability that p is not incident to any edge in E is at most $\prod_{pq\in E_P}(1 - x_{pq}) \leq \exp(-\sum_{pq\in E_P} x_{pq}) \leq 1/e$. Thus, letting μ denote the expected number of singletons, the expected number of components in (P,E) is at most $(n-\mu)/2 + \mu = n/2 + \mu/2 \leq n(1/2 + 1/2e)$. By Markov's inequality, with probability at least 1/20, the graph (P,E) has at most $3n/4$ components.

Consider a line $\ell \in L_P$. By Chernoff's bound, the probability that ℓ crosses more than $2e\sqrt{n}$ edges in E is at most $2^{-2e\sqrt{n}}$. Taking a union bound, the probability that any line in L_P crosses more than $2e\sqrt{n}$ segments is much less than 1/30, for n large enough. Thus, E fulfills the claimed properties with positive probability, which implies existence. ☐

Theorem 5 *Let P be a planar n-point set in general position. There exists a spanning tree T for P with crossing number $d\sqrt{n}$, for some fixed $d > 0$.*

Proof. We use induction on n. For $n = O(1)$, the statement holds. For larger n, use Lemma 4 to obtain a set $E \subseteq E_P$ such that (P,E) has at most $3n/4$ components and crossing number $c\sqrt{n}$. Let P' contain one vertex from each connected component of (P,E). By induction, P' has a spanning tree T' with crossing number at most $d\sqrt{|P'|} \leq d\sqrt{3n/4}$. The union $T' \cup E$ is a spanning graph for P with crossing number at most $c\sqrt{n} + d\sqrt{3n/4} \leq d\sqrt{n}$, for d large enough. We take a spanning tree of this graph. ☐

3 Heuristics and Experiments

We present a new simple heuristic, the *Connected Components* algorithm, for finding a spanning tree with low crossing number. We have implemented it and three other algorithms. First, we briefly describe the methods, and then we report on the experiments. In the following, L is an arbitrary (finite) set of lines.

Iterative Reweighting [5, 11]. We follow the presentation in [9]. The algorithm constructs a spanning tree by adding the edges one by one. Let $E_i \subset E_P$ be the edges after the ith iteration (with $E_0 = \emptyset$). Each line $\ell \in L$ is assigned a weight, which at the beginning of the ith iteration is $w_{i-1}(\ell) = 2^{n_{i-1}(\ell)}$, where $n_{i-1}(\ell) = |\{e \in E_{i-1} \mid e \cap \ell \neq \emptyset\}|$. Accordingly, the weight of an edge e is

$$w_{i-1}(e) = \sum_{\ell\in L:\, \ell\cap e\neq\emptyset} w_{i-1}(\ell).$$

In iteration i, the algorithm selects a lightest edge between two different components in (P, E_{i-1}). This is repeated until a spanning tree has been formed.

Har-Peled's algorithm ([8], see also [4, 6]) This is an implementation of the proof of Theorem 5: we set up an LP as in the beginning of Section 2, replacing $\sqrt{n}$ by a parameter t to be optimized. We solve the LP and use the resulting variables x_{pq} as probabilities to sample a subset of edges E. With constant probability, (P,E) has at most $3n/4$ components, in which case we take one point from each component (otherwise we resample). We repeat until the problem size is $O(1)$, and we return a spanning tree from the union of all sampled edges. One can show that the result has expected crossing number $O(t^* \log n)$, where t^* is the optimum.

Iterative LP-rounding [7]. Fekete et al. gave the following IP for computing an optimum spanning tree:

$$\begin{aligned} &\text{minimize } t \\ &\text{s. t.} \sum_{pq\in E_P} x_{pq} = n - 1 \\ &\sum_{pq\in\delta(S)} x_{pq} \geq 1 \qquad \forall\, \emptyset \neq S \subset P \qquad (*) \end{aligned}$$

$$\sum_{pq \in E_P :\, pq \cap \ell \neq \emptyset} x_{pq} \leq t \qquad \forall\, \ell \in L$$

$$x_{pq} \in \{0,1\} \qquad \forall\, pq \in E_P,$$

where $\delta(S) := \{pq \in E_P \mid p \in S, q \notin S\}$ is the cut induced by S. They also suggested the following iterative rounding algorithm: repeatedly solve the relaxed LP (where $0 \leq x_{pq} \leq 1$), each time fixing the value of a heaviest edge to one (by adding a constraint to the LP). Despite the exponential number of *blossom constraints* (*), Fekete et al. suggested that the LP can be solved in polynomial time by using a separation oracle. Unfortunately, we were unable to implement such an oracle in our LP solver (Gurobi; see also below). Moreover, a heaviest edge may render the LP infeasible by creating a cycle. As a remedy, we opted for the following polynomially many *fractional connectivity* constraints

$$\sum_{pq \in E_P :\, p \in C, q \notin C} x_{pq} \geq 1 \quad \forall\, C \in \mathcal{C},$$

where $\mathcal{C}$ is the set of the connected components induced by the edges selected so far. The constraints ensure that each component has at least one outgoing edge. Finally, a heaviest edge between two components is selected.

Connected Components. Our approach is similar to the Iterative LP-rounding above, but now the LP models only the edges among the connected components in $\mathcal{C}$. Let $E(\mathcal{C})$ be these edges. We have

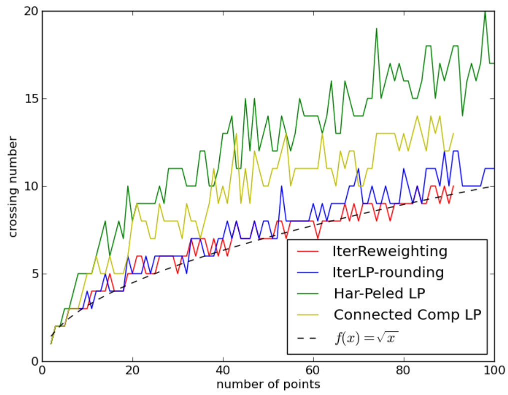
$$\begin{aligned} &\text{minimize } t \\ &\text{s. t.} \sum_{pq \in E(\mathcal{C})} x_{pq} = |\mathcal{C}| - 1 \\ &\sum_{pq \in E(\mathcal{C}) :\, p \in C} x_{pq} \geq 1 && \forall C \in \mathcal{C} \\ &\sum_{pq \in E(\mathcal{C}) :\, pq \cap \ell \neq \emptyset} x_{pq} \leq t && \forall \ell \in L \\ &x_{pq} \geq 0 && \forall pq \in E(\mathcal{C}) \end{aligned}$$

At each iteration the heaviest edge is selected, and the algorithm runs until there is only one component.

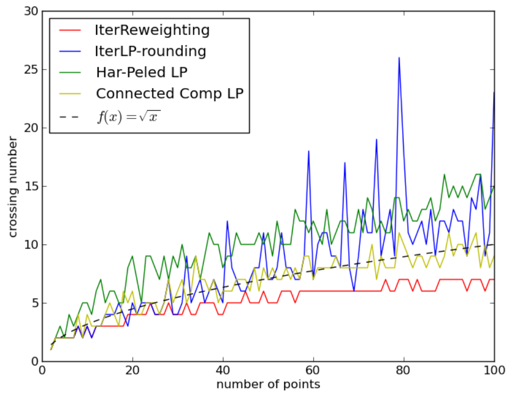
3.1 Experimental Results

We ran the algorithms on artificial data and on real TSP instances from TSPLIB [2]. The experiments were run on a GNU/Linux Debian Wheezy server with eight Intel Xeon E 5440 CPUs at 2.83 GHz and 32 GB of shared RAM. The code was written in Python (v.2.7.3), while for solving LPs we used Gurobi (v.5) with four threads [1].

Artificial Data. These consist of points sampled uniformly at random from the integer $[n] \times [n]$ grid and randomly perturbed by some small ϵ, with (a) all $\Theta(n^2)$ lines, i.e., $L = L_P$ and (b) $\Theta(\sqrt{n})$ random lines; see Fig. 1. In (a) all algorithms produce a spanning tree with a crossing number of $O(\sqrt{n})$, with Iterative Reweighting performing best. In (b) Iterative Reweighting yields a crossing number that is noticeably lower than $O(\sqrt{n})$.



(a) All lines



(b) Random lines

Figure 1: Results on random points.

Fig. 2 shows the average crossing number (i.e., number of all crossings over the number of lines) on random points and random lines chosen as before. Iterative Reweighting and Connected Components yield the best results and produce an average crossing number of $O(\log n)$.

TSPLIB. The results are shown in Fig. 3. For small instances, the crossing number was computed w.r.t. all $\Theta(n^2)$ lines, while for large instances we used random lines since, computationally, the 'all lines' case proved to be prohibitively expensive. Iterative Reweighting produces a spanning tree with lowest crossing number for all instances except for the ulysses ones, for which Iterative LP-rounding wins. Har-Peled's algorithm and Iterative LP-rounding are fastest, while Iterative Reweighting and Connected

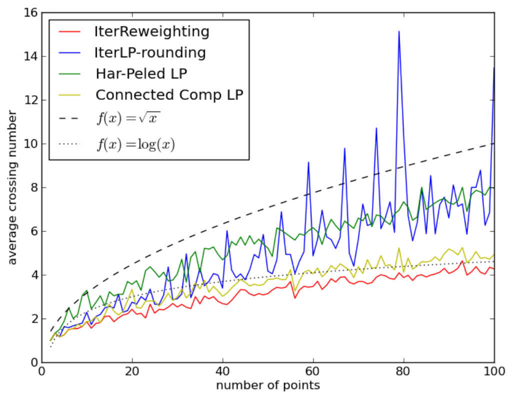


Figure 2: Average crossing number on random points with random lines.

Data set	Lines |L|	Lines type	IterReweighting ✂	IterReweighting ✂̄	IterReweighting CPU	IterLP-rouding ✂	IterLP-rouding ✂̄	IterLP-rouding CPU	Har-PeledLP ✂	Har-PeledLP ✂̄	Har-PeledLP CPU	ConnectCompLP ✂	ConnectCompLP ✂̄	ConnectCompLP CPU
berlin52	1036	all	7	3.97	454.60	8	4.75	199.08	16	6.71	208.52	10	5.17	503.61
bier127	120	rand.	5	1.94	491.99	20	6.87	142.85	16	7.52	137.15	8	2.85	639.01
eil51	1043	all	6	2.91	419.33	11	4.91	187.10	10	6.14	198.38	9	3.43	478.28
eil76	2198	all	6	2.64	2377.09	15	7.15	896.56	16	7.89	922.36	9	3.40	3020.36
eil101	138	rand.	4	1.73	313.66	15	6.94	103.31	15	7.30	99.99	8	2.43	389.78
lin105	93	rand.	4	2.14	245.24	22	8.60	75.37	11	5.11	73.09	5	2.62	293.81
u159	129	rand.	2	0.69	1109.36	24	12.62	254.18	17	6.41	231.49	3	0.87	1450.29
ulysses16	122	all	5	2.70	2.65	4	2.92	2.08	8	3.87	2.36	7	3.25	2.82
ulysses22	208	all	6	3.07	10.84	5	3.83	6.86	11	4.50	7.77	7	3.54	10.90

Figure 3: Results on TSPLIB instances; ✂: crossing number, ✂̄: average crossing number, CPU: computing time in seconds. The table is split in three parts, the data set and line number/type appear only in the first part.

Components achieve relatively similar results w.r.t. the crossing and average crossing number.

We have also computed an optimal solution for ulysses16 using the ILP by Fekete et al., see Fig. 4. Iterative Reweighting and Iterative LP-rounding get closest to the optimal crossing numbers. Due to the exponential number of the blossom inequalities in the ILP we were not able to compute optimal solutions for larger instances within reasonable processing time.

Algorithm	✂	✂̄	CPU
OPT	4	2.84	85.00
IterReweighting	5	2.70	2.65
IterLP-rounding	4	2.92	2.08
Har-PeledLP	8	3.87	2.36
ConnectedCompLP	7	3.25	2.82

Figure 4: Comparison to an optimal solution on ulysses16 with $L = L_P$, $\mid L \mid= 122$.

Acknowledgments. We would like to thank Nabil Mustafa, Sariel Har-Peled and Sandor Fekete for helpful discussions.

References

[1] Gurobi Optimizer v.5. `http://www.gurobi.com/`.

[2] TSPLIB. `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`.

[3] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

[4] V. Bilò, V. Goyal, R. Ravi, and M. Singh. On the crossing spanning tree problem. In *Proc. 8th/7th APPROX-RANDOM*, pages 51–60, 2004.

[5] B. Chazelle and E. Welzl. Quasi-optimal range searching in space of finite VC-dimension. *Discrete Comput. Geom.*, 4:467–489, 1989.

[6] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding for matroid polytopes and applications. `arXiv:0909.4348`, 2009.

[7] S. P. Fekete, M. E. Lübbecke, and H. Meijer. Minimizing the stabbing number of matchings, trees, and triangulations. *Discrete Comput. Geom.*, 40(4):595–621, 2008.

[8] S. Har-Peled. Approximating spanning trees with low crossing number. `arXiv:0907.1131`, 2009.

[9] S. Har-Peled. *Geometric approximation algorithms.* AMS, 2011.

[10] J. Matoušek. *Lectures on discrete geometry.* Springer-Verlag, New York, 2002.

[11] E. Welzl. On spanning trees with low crossing numbers. In *Data Structures and Efficient Algorithms, LNCS 594*, pages 233–249, 1992.