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 \( E_\ell \subseteq E \) be the set of all edges that intersect \( \ell \). Conversely, for \( pq \in E \), 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 \( r \in \{0, \ldots, \lfloor |L|/2 \rfloor\} \), the disk \( D_L(p, r) \) contains at least \( \left( \frac{r+1}{2} \right) \) 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 \( A^Ty \geq 0, b^Ty < 0, y \geq 0 \).

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.

\[
\begin{align*}
\sum_{pq \in E_\ell} x_{pq} & \leq \sqrt{n}, \text{ for all } \ell \in L_P \\
\sum_{pq \in E_P} x_{pq} & \geq 1, \text{ for all } p \in P \\
x_{pq} & \geq 0, \text{ for all } pq \in E_P.
\end{align*}
\]

Lemma 3 The LP is feasible.

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

\[
\sqrt{n} \sum_{\ell \in L_P} y_{\ell} < \sum_{p \in P} y_p \quad (*)
\]

\[
\sum_{\ell \in L_{pq}} y_{\ell} \geq y_p + y_q, \text{ for all } pq \in E_P \\
y_{\ell} \geq 0, y_p \geq 0, \text{ for all } \ell \in L_P, p \in P.
\]

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

\[
\sqrt{n} \sum_{\ell \in L_P} \lambda y_{\ell} = \sum_{p \in P} \lambda y_p - \lambda c
\]

Fix \( \lambda \in \mathbb{N} \) and set \( z = \lambda y \). For \( \lambda \) large enough, \( z \) is integral. Let \( L \) be the line set with \( z_{\ell} \) copies of \( \ell \), for all \( \ell \in L_P \), 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 $d_L(p, q) = \sum_{\ell \in L_p} z_\ell \geq z_p + z_q$. 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 $\sum_{pq \in E} (\frac{z}{2}) \leq (\frac{N}{2})$, since the arrangement of $L$ has $(\frac{N}{2})$ vertices. Hence,

$$\sum_{p \in P} z_p \leq \sqrt{n} \sum_{p \in P} \frac{z^2_p}{2} \leq \sqrt{n} \sum_{p \in P} 2 \left(\frac{z_p}{2}\right)^2 + z_p \leq \sqrt{n} \sum_{p \in P} \left(\frac{\sqrt{n}}{2}\right) + z_p \leq \sum_{p \in P} \lambda y_p - \sum_{p \in P} \lambda y_p,$$

where we used Cauchy-Schwarz; $a^2 = 2\left(\frac{n}{2}\right) + a$; 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, $\lambda \leq (n/c^2) \sum_{p \in E_P} y_p$, a contradiction to $\lambda \in \mathbb{N}$ being arbitrary. Hence, the LP must be feasible. $\square$

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

\textbf{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 $c > 0$.

\textbf{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 $\mu$ 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 $\ell$ 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. $\square$

\textbf{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 $d > 0$.

\textbf{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 $d\sqrt{3n/4}$. 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. $\square$

\section{Heuristics and Experiments}

We present a new simple heuristic, the \textit{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.

\textbf{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 $i$th iteration (with $E_0 = \emptyset$). Each line $\ell \in L$ is assigned a weight, which at the beginning of the $i$th 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

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

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.

\textbf{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.

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

\begin{align*}
\text{minimize} & \quad t \\
\text{s. t.} & \quad \sum_{pq \in E} x_{pq} = n - 1 \\
& \quad \sum_{pq \in E(S)} x_{pq} \geq 1 \quad \forall \emptyset \neq S \subset P \\
& \quad (*)
\end{align*}
\[
\sum_{pq \in E_P : \ pq \cap \ell \neq \emptyset} x_{pq} \leq t \quad \forall \ell \in L
\]
\[
x_{pq} \in \{0,1\} \quad \forall \ pq \in E_P,
\]

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
\[
\sum_{pq \in E_P : \ pq \cap \ell \neq \emptyset} x_{pq} \leq t \quad \forall \ell \in L
\]
\[
x_{pq} \geq 0 \quad \forall \ pq \in E_P,
\]
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 \( O(\sqrt{n}) \).

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 ulyses ones, for which Iterative LP-rounding wins. Har-Peled’s algorithm and Iterative LP-rounding are fastest, while Iterative Reweighting and Connected Components.
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.

We have also computed an optimal solution for ulysses16 using the ILP by Fekete et al., see Fig. 4. Components achieve relatively similar results w.r.t. the crossing and average crossing number.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Lines</th>
<th>type</th>
<th>IterReweighting</th>
<th></th>
<th></th>
<th></th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>berlin52</td>
<td>1036</td>
<td>all</td>
<td>7</td>
<td>3.97</td>
<td>454.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bier127</td>
<td>120</td>
<td>rand.</td>
<td>5</td>
<td>1.94</td>
<td>491.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ell51</td>
<td>1043</td>
<td>all</td>
<td>6</td>
<td>2.91</td>
<td>419.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ell76</td>
<td>2198</td>
<td>all</td>
<td>6</td>
<td>2.64</td>
<td>2377.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ell101</td>
<td>138</td>
<td>rand.</td>
<td>4</td>
<td>1.73</td>
<td>313.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lin105</td>
<td>93</td>
<td>rand.</td>
<td>4</td>
<td>2.14</td>
<td>245.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>u159</td>
<td>129</td>
<td>rand.</td>
<td>2</td>
<td>0.69</td>
<td>1109.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ulysses16</td>
<td>122</td>
<td>all</td>
<td>5</td>
<td>2.70</td>
<td>2.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ulysses22</td>
<td>208</td>
<td>all</td>
<td>6</td>
<td>3.07</td>
<td>10.84</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IterLP-rounding</th>
<th></th>
<th></th>
<th>Har-PeleLP</th>
<th></th>
<th></th>
<th></th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4.75</td>
<td>199.08</td>
<td>16</td>
<td>6.71</td>
<td>208.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>6.87</td>
<td>142.85</td>
<td>16</td>
<td>7.52</td>
<td>137.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4.91</td>
<td>187.10</td>
<td>10</td>
<td>6.14</td>
<td>198.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>7.15</td>
<td>896.56</td>
<td>16</td>
<td>7.89</td>
<td>222.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>6.94</td>
<td>103.31</td>
<td>15</td>
<td>7.30</td>
<td>99.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>8.60</td>
<td>75.37</td>
<td>11</td>
<td>5.11</td>
<td>73.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>12.62</td>
<td>254.18</td>
<td>17</td>
<td>6.41</td>
<td>231.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.92</td>
<td>2.08</td>
<td>8</td>
<td>3.87</td>
<td>2.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.83</td>
<td>6.86</td>
<td>10</td>
<td>4.50</td>
<td>7.77</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ConnectCompLP</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5.17</td>
<td>503.61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.85</td>
<td>639.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3.43</td>
<td>478.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3.40</td>
<td>3020.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.43</td>
<td>389.78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.62</td>
<td>293.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.87</td>
<td>1450.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3.25</td>
<td>2.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3.54</td>
<td>10.90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Results on TSPLIB instances; $\times$: crossing number, $\Rightarrow$: 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.

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

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

References