A quadratic distance bound on sliding between crossing-free spanning trees ⋆

Oswin Aichholzer [a,*,1] Klaus Reinhardt [b]

[a] *Institute for Software Technology, Graz University of Technology, Inffeldgasse 16b, A-8010 Graz, Austria,* oaich@ist.tugraz.at

[b] *Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Sand 13, D-72076 Tübingen,* reinhard@informatik.uni-tuebingen.de

Abstract

Let S be a set of n points in the plane and let $\mathcal{T}_S$ be the set of all crossing-free spanning trees of S. We show that it is possible to transform any two trees in $\mathcal{T}_S$ into each other by $O(n^2)$ local and constant-size edge slide operations. Previously no polynomial upper bound for this task was known, but in [1] a bound of $O(n^2 \log n)$ operations was conjectured.

Key words: Crossing free spanning trees, transformation, Flips, Edge slides

1 Introduction

Let S be a set of n points in the Euclidean plane. A *crossing-free spanning tree* of S is a tree whose edges connect all points in S (and no others) with straight line segments that pairwise do not cross. With $\mathcal{T}_S$ we denote the set of all crossing-free spanning trees of S.

An interesting question is whether, and how fast, two members of $\mathcal{T}_S$ can be transformed into each other by means of predefined rules, often called flips. A common operation is what is called an *edge move*, which relates two trees in the set $\mathcal{T}_S$ iff they have all but one edge in common (one edge is 'flipped'). For this general setting Avis and Fukuda [2] showed that the corresponding tree graph is connected and has a diameter bounded by $2n-4$. If we restrict the set

⋆ A preliminary version of this paper appeared as [3].

* Corresponding author.

[1] Research supported by Acciones Integradas 2003-2004, Proj.Nr.1/2003.

of allowed flips to length-improving edge moves then in [1] a way to transform any tree $T \in \mathcal{T}_S$ into the minimum spanning tree of S in only $O(n \log n)$ steps was given. For a more detailed discussion of this subject and some historical background see [1].

Our interest is focused on a local edge move that keeps one endpoint of the moved edge fixed and moves the other one along an adjacent tree edge. Following [4], we call this constant-size operation an edge slide. An edge slide can be carried out in a (geometrically) continuous way. Thus, together with the results given below, it allows for a continuous deformation of crossing-free spanning trees into each other. More formally the operation we consider is defined as follows [1]: Consider a tree $T' \in \mathcal{T}_S$. A *(planar) edge slide* on T' takes some edge $e \in T'$ and moves one of its endpoints along some edge adjacent to e in T', without generating any edge crossings. This gives a new edge f and a new tree $T'' = T' \cup \{f\} \setminus \{e\}$ such that $T'' \in \mathcal{T}_S$. An edge slide is a special kind of planar edge move: T'' is obtained by closing with f a 3-cycle C in T' and by removing e from C, such that T' avoids the interior of the triangle C. To put it differently: If two adjacent edges ab and ac of T' form a triangle abc that does not intersect T', then either of ab or ac can be changed to bc. Intuitively speaking, an edge slide is an edge operation as local as it can be.

In this paper we investigate the questions of how fast two crossing-free spanning trees of $\mathcal{T}_S$ can be transformed into each other by means of the edge slide operation. To this end consider the *tree graph* $\mathcal{TG}(S)$ which is an undirected graph that has $\mathcal{T}_S$ as its set of nodes. It realizes an arc between two nodes (trees) T' and T'' if and only if T' can be transformed into T'' by an edge slide (and vice versa). In [1] it was shown that $\mathcal{TG}(S)$ is connected. This implies transformability of the corresponding trees into each other by a local and constant-size operation, which was the first result in this direction. The length of a shortest path in $\mathcal{TG}(S)$ corresponds to the distance between the two respective trees. However, for the edge slide operation no polynomial upper bound on this length has been known. It was conjectured that 'if two trees are part of the same triangulation of S then they can be transformed into each other by $O(n^2)$ edge slides'. By results in [1], this would give a diameter of $O(n^2 \log n)$ for the corresponding tree graph.

We are able to prove the following, slightly stronger result:

Theorem 1 *Let T' and T'' be any two crossing-free spanning trees of S. Then T' can be transformed into T'' by $O(n^2)$ edge slides.*

As mentioned in [1] the edge slide operation could also prove useful in enumerating all simple polygons on a point set S via constant-size local transformations. This question is still unsettled; see e.g. Hernando et al. [6]. Our upper bound on the diameter of $\mathcal{TG}(S)$ might be useful in this respect.

2 Upper bound Construction

Let S and $T \in \mathcal{T}_S$ be as defined in Section 1. W.l.o.g. we assume that no two points of S have the same x-coordinate (otherwise we rotate the coordinate system appropriately). We call a pair (e, p_j), where $e = (p_i, p_k)$ is an edge of T and $p_i, p_j, p_k \in S$ are sorted in x-order, a *slide triangle* if the open triangle $\Delta = p_i, p_j, p_k$ is free of points from S and edges from T, that is, the interior of Δ is empty.

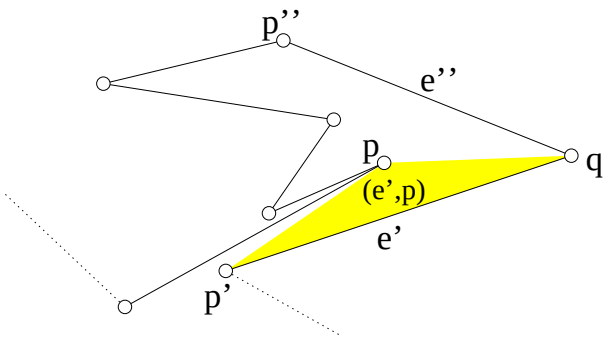

Fig. 1. Illustration of the proof of Lemma 1

Lemma 1 *Let P be a simple polygon with vertex set S, and let δP be the boundary of P with one marked edge e^*. If $\delta P \setminus \{e^*\}$ is not an x-monotonous path then in the interior of P there always exists a slide triangle $(e, p) \subset P$, $e \neq e^*$.*

Proof. Since $\delta P \setminus \{e^*\}$ is not x-monotonous there exists a vertex $q \in S$ with two edges from $\delta P \setminus \{e^*\}$ both emanating to the same side, i.e., both to the left or right of q. Let e' and e'', respectively, be these edges. W.l.o.g. we assume that they emanate from q to the left, e' lies below e'' and the left endpoint p' of e' lies to the left of the left endpoint $p''$of e'', see Figure 1 (all other cases are symmetric). If the open triangle $\Delta = q, p', p''$ is empty we have a slide triangle (e', p''). Otherwise consider the point p which, among all points of P in the interior of Δ, minimizes the angle $\angle pp'q$ at p'. Note that there are no edges (partially) inside Δ that have q as an endpoint or intersect e' or e''. Therefore p provides a slide triangle (e', p). □

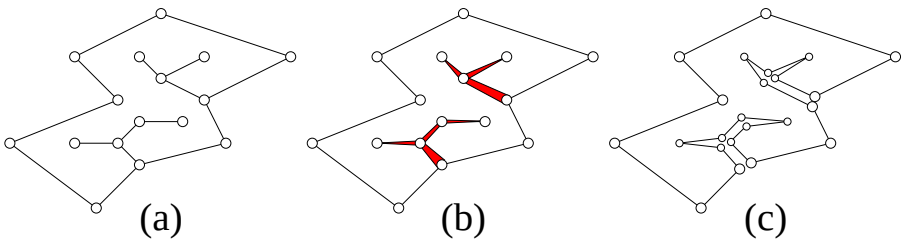

Fig. 2. Cutting a tree polygon (a) along interior edges (b) to obtain a simple polygon (c).

A *tree polygon* P is a simple polygon with interior points, each point connected to the boundary δP via a unique (simple) path such that the resulting graph is crossing-free, see Figure 2(a). In other words, the graph without the edges of δP is a forest. We claim that we can handle this more general situation like a simple polygon: Cut along interior edges and move them apart at the cuts infinitesimally, i.e., duplicate the related vertices, see Figure 2(b) and (c). Observe that the proof of Lemma 1 still holds for this setting by considering edges e' and e'' that are neighboring in the cyclic order around q.

We call the x-monotonous path connecting all vertices of S in their x-sorted order the *canonical spanning tree* $T_c \in \mathcal{T}_S$ of S.

Theorem 2 *For a point set S and a crossing-free spanning tree $T \in \mathcal{T}_S$, $T \neq T_c$, there always exists a slide triangle (e, p), $p \in S$ and $e \in T$ such that the path $\pi \in T$ connecting p to e, say at point q, is x-monotonous. Moreover $\pi \cup pq$ is a simple polygon without interior points.*

Proof. We first show that there always exists some slide triangle (e, p). The union of T and the boundary of the convex hull of S partitions S into $k \geq 1$ tree polygons P_i, $i = 1, \ldots, k$. Since T is a connected spanning tree each P_i has a unique boundary edge not belonging to T. We mark these edges. From Lemma 1 and the discussion afterwards we know that we get a slide triangle inside some P_i unless for all P_i the remaining (non marked) part is x-monotonous. But in the latter case T must be an x-monotonous path, that is, $T = T_c$, a contradiction.

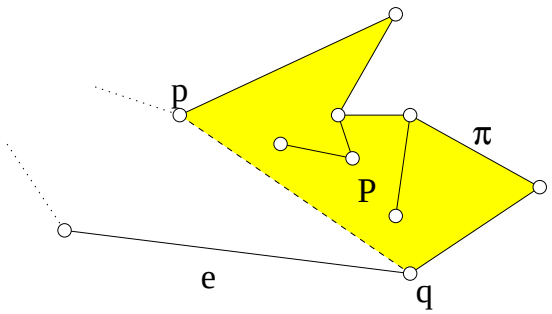

Fig. 3. Illustration of the proof of Theorem 2

Let q be the (first) endpoint of e to which p is connected. If the edge pq belongs to T we are done. Thus assume that p is connected to q via a path π of length greater than 1, see Figure 3. Since (e, p) is a slide triangle the edge p, q does not cross an edge of T. Thus the 'pocket' formed by π together with the edge p, q and possible interior edges and points is a tree polygon P. If $\delta P \setminus \{pq\}$ is an x-monotonous path we are done. Otherwise we mark the edge p, q and iterate recursively on P. Note that only one edge of P is marked, since T does not contain cycles. Moreover, in every iteration we obtain a smaller instance, since we get rid of at least one edge of T. $\square$

For an edge e its weight is defined as the number of points from S which lie in the open x-interval spanned by e, that is, the number of points which lie between the endpoints of e in the x-sorted order. The weight of a tree T, denoted by $w(T)$, is the sum of the weights of its edges. Obviously T_c has weight zero and is the only tree with this property. Since each of the $n-1$ edges of T has at most weight $n-2$ the weight of a tree with n points is bounded by $(n-1)(n-2) < n^2$. A tight bound is given by the following Lemma.

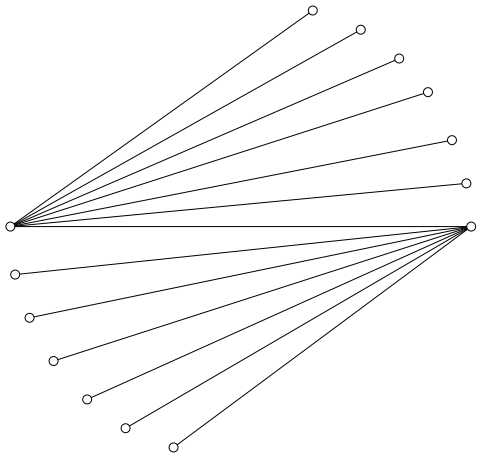

Fig. 4. An tree with maximum weight of $\frac{3n^2-10n+8}{4}$.

Lemma 2 *The weight $w(T)$ of a crossing-free spanning tree $T \in \mathcal{T}_S$ is bounded by $0 \leq w(T) \leq \lfloor \frac{3n^2-10n+8}{4} \rfloor$, $n \geq 2$, and these bounds are tight.*

Proof. Let the leftmost point be the root $r \in S$ of T. Associate each vertex of T except r to its unique incident edge lying on the path towards r. Thus in this setting we can argue about the weight of the vertices instead of the weight of the edges.

To maximize the weight of a tree consider the x-sorted sequence of the vertices. A vertex maximizes its weight if its associated edge connects it to either the leftmost or rightmost vertex of the set. In other words the i-th vertex gains a weight of up to $\max\{i-2, n-i-1\}$. For the whole tree T we thus get $w(T) \leq \sum_{i=2}^{n} \max\{i-2, n-i-1\}$. This gives $w(T) \leq \frac{3n^2-10n+8}{4}$ for n even and $w(T) \leq \frac{3n^2-10n+7}{4}$ for n odd, respectively. From the argumentation it is clear that there exist point sets with crossing-free spanning trees obtaining these bounds, see Figure 4 for a generic example. □

In fact the above proof does not require the tree to be crossing-free, i.e., Lemma 2 holds for all spanning trees of S.

Lemma 3 *Any crossing-free spanning tree $T \in \mathcal{T}_S$ can be transformed into T_c by at most $2 \cdot w(T)$ edge slides.*

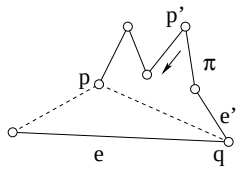

Fig. 5. Slide triangle (e, p) with x-monotonous path π connecting p to q.

Proof. If $T = T_c$ the statement is obviously true, so let $T \neq T_c$. Let (e, p) be a slide triangle as provided by Theorem 2, see Figure 5. Let $k \geq 1$ be the number of edges of the x-monotonous path π connecting p to some endpoint q of e. We claim that we can reduce the weight of e by at least k by performing $2k - 1$ edge slides. To this end let e' be the edge of π incident to q. Our first task is to slide e' along π to obtain the edge qp.

Assume that $k > 1$. Then π avoids the interior of the slide triangle (e, p) and thus contains at least one vertex p' pointed away from the edge qp. Since π is x-monotonous we can slide the edge of π which has p' as its left endpoint 'towards' p along the edge of π which has p' as its right endpoint. We iterate this process until we obtain the edge qp, i.e., $k = 1$. Since each edge slide reduces the length of the current path from p to q by one, we carry out exactly $k - 1$ steps.

Now we can slide e along qp, reducing its weight by at least k (the vertices of π different from q). Finally we slide qp back to e' by reversing the steps of the first phase.

As long as the resulting tree is not T_c we repeat all above steps. After each iteration the weight of a single edge has been decreased by at least half of the number of the involved edge slide operations. We thus can transform T into T_c with at most $2w(T)$ edge slides. □

We are now ready to prove our main result as proposed in Section 1. We give here a more explicit statement and Theorem 1 then follows as a corollary.

Theorem 3 *For any pair $T', T'' \in \mathcal{T}_S$ we can transform T' into T'' by at most $2(w(T') + w(T'')) \leq 3n^2$ edge slides.*

Proof. Lemma 3 shows that we can transform any tree $T' \in \mathcal{T}_S$ into T_c with at most $2w(T')$ edge slides. By symmetry of the edge slide operation we can use the reverse transformation for T''. Together with the upper bound $w(T) \leq \frac{3n^2}{4}$ from Lemma 2 for any $T \in \mathcal{T}_S$, the theorem follows. □

Java-applets demonstrating an algorithmic interpretation can be found under `http://www-fs.informatik.uni-tuebingen.de/~reinhard/esa.html` and `http://www.ist.tugraz.at/staff/aichholzer/treeslides/Welcome.html`

3 Lower bound Construction

The following theorem shows that there are examples requiring $\Omega(n^2)$ edge slides to transform two spanning trees into each other. Thus the bound of Theorem 1 is asymptotically tight.

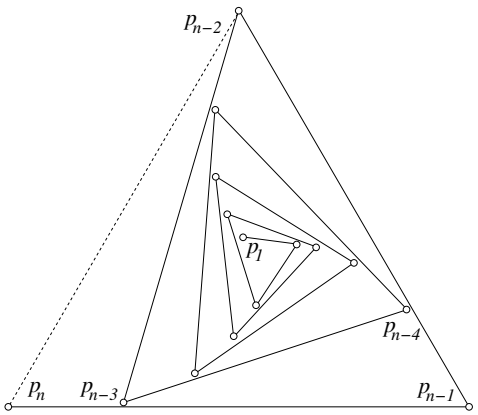

Fig. 6. An example requiring $\Omega(n^2)$ edge slides.

Theorem 4 *Let $T = \{p_1 - p_2 - ... - p_n\}$ for odd $n > 2$ have the shape of a spiral like in Figure 6 labeled from inside to outside. Then we need at least $(n-1)(n-2)/2$ edge slides to get a tree containing the edge (p_{n-2}, p_n).*

Proof. We prove by induction on odd k that we need at least $(k-1)(k-2)/2$ edge slides to move any edge being or becoming adjacent to a point p_j with $j \geq k$ for any $2 < k \leq n$. For $k = 3$ we need $(3-1) \cdot (3-2)/2 = 1$ slide.

For odd $k > 3$ we first need to move an edge to (p_{k-3}, p_{k-1}) before we can move any edge being or becoming adjacent to a point p_i with $i \geq k$ since (p_{k-1}, p_k) is the only connection of p_k to a point with lower index (and thus must be slided or slided along) and p_{k-3} is the only point with lower index than k having an empty triangle with (p_{k-1}, p_k) that would allow the slide. (The point p_{k-3} is inside the triangle (p_{k-4}, p_{k-1}, p_k)). With the same argument, we first need the edge (p_{k-4}, p_{k-2}) before we can move any edge being or becoming adjacent to a point p_i with $i \geq k-1$.

By induction we need at least $(k-3)(k-4)/2$ edge slides to get (p_{k-4}, p_{k-2}). At this moment (p_{k-2}, p_{k-1}) and (p_{k-1}, p_k) are still there. Now we make use of the locality of the slide operation in combination with the fact that there is exactly one path between two nodes in a tree and consider the following two numbers: the number q (initially 1) of nodes (initially p_{k-1}) on the path from p_{k-2} to p_k and the number i (initially $k-3$) of internal nodes of the tree polygon formed by the edge (p_{k-2}, p_k) and the tree (these are the nodes inside the polygon formed by the path from p_{k-2} to p_k and (p_{k-2}, p_k)). Both numbers q and i have to be 0 when the edge (p_{k-2}, p_k) is reached. Sliding an

edge, which is not part of the path has no effect on q or i (and does not help to get (p_{k-2}, p_k)). Sliding an edge, which is part of the path along another edge of the path decreases q by one but can not decrease i since the slide-triangle (which was cut away from the polygon) must be empty and internal nodes stay internal nodes. Sliding an edge, which is part of the path along an edge not being part of the path may decrease i by 1 (if it was connecting to an internal node) but also increases q by 1 (An internal node became a node on the path and a triangle was cut away from the polygon). This means each of the following moves leading to (p_{k-2}, p_k) decreases the parameter $q+2i$ which is initially $1+2(k-3) = 2k-5$ by at most 1. Thus the total number of moves is at least $(k-3)(k-4)/2 + 2k - 5 = (k-1)(k-2)/2$. □

The previous Theorem 4 does not restrict the rest of the resulting tree except the edge (p_{n-2}, p_n). We can improve the lower bound by a factor of 2 demanding the spiral structure in the resulting tree as well:

Theorem 5 *Let $T_1 = \{p_1 - p_2 - ... - p_{n+1}\}$ for odd $n > 2$ have the shape of a spiral like in Figure 6 numbered from inside to outside. Let T_2 be the tree where (p_{n+1}, p_n) in T_1 is replaced by (p_{n+1}, p_{n-2}). Furthermore we assume that p_{n+1} is so close to p_{n-2} that no straight line spanned by two other points of S separates them. Then we need at least $(n-1)(n-2)+1$ edge slides to transform T_1 into T_2.*

Proof. With the same argument as in the proof of Theorem 4 we need the edge (p_{n-2}, p_n) at some stage on the way from T_1 to T_2. For the number $(n-1)(n-2)/2$ of slides from T_1 to a tree containing the edge (p_{n-2}, p_n), we can apply Theorem 4 directly. For the same number $(n-1)(n-2)/2$ of slides from T_2 to a tree containing the edge (p_{n-2}, p_n), we can use a slight variation of the proof for Theorem 4: Each time the point p_{n-2} is considered, alternative connections to p_{n+1} are considered too, but since both points are so close, there is no difference between both cases. Furthermore one additional slide from (p_{n+1}, p_n) to (p_{n+1}, p_{n-2}) is required, which results in at least $(n-1)(n-2)+1$ slides in total. □

From this spiral example we can not obtain a better lower bound, because a sequence of exactly the given number of slides indeed exists. Consider the stage in the proof of Theorem 4 where we have the edges (p_{k-4}, p_{k-2}), (p_{k-2}, p_{k-1}) and (p_{k-1}, p_k). We start sliding (p_{k-2}, p_{k-1}) to (p_{k-4}, p_{k-1}), then we reverse the $2(k-2)-5$ slides from the recursion which reestablishes the edges (p_{k-2}, p_{k-3}) and (p_{k-3}, p_{k-4}) then we slide (p_{k-4}, p_{k-1}) to (p_{k-3}, p_{k-1}) to (p_{k-3}, p_k) and finally to (p_{k-2}, p_k). All together $1 + 2(k-2) - 5 + 3 = 2k - 5$ slides.

One might wonder whether the slide-distance between two spanning trees which do not intersect each other is smaller than in the general case. A result of this type holds for triangulations, where the flip-distance can be bounded by the number of crossing edges [5]. However, from the example in Figure 6 it follows that even for two trees differing in only one edge the slide-distance is quadratic.

Another observation is that the weight of a spanning-tree is direction-sensitive. So an obvious question is whether there always exists a 'nice' direction with sub-quadratic weight? Again a negative answer is given by the example of Figure 6, having weight $\Theta(n^2)$ for any direction of the x-axis. However this would still give a chance to improve the constant factor of the upper bound and minimize the gap between our upper and lower bound, which is currently a factor of 3.

A related algorithmic question is how fast we can compute a direction to minimize the weight of a given tree. This can be done in time $O(n^2 \log n)$ as follows. First cyclically sort all directions induced by pairs of points of S (in time $O(n^2 \log n)$). Then calculating the x-order of the points and the total weight ($O(n^2)$ time). Continue turning the direction by swapping two neighbor points (w.r.t. sorting) while locally changing the weight. Since the average degree of the points is constant, the cost is dominated by the $O(\log n)$ factor for the operations on the priority queue and the total time is $O(n^2 \log n)$. (We can even keep the required space linear if instead of sorting all directions, we only keep current neighbor points in a priority queue according to their direction. The priority queue delivers the next pair to swap.)

The presented results are of combinatorial nature, i.e., bound the number of necessary slide operations. On the algorithmic side one might be also interested in the time complexity to compute the $O(n^2)$ slide sequence. It is not hard to see from our construction, that it can be done in polynomial time but we leave an exact estimation as an open problem for future investigations.

Related is the question on how fast we can check whether a particular edge slide is valid since it must not intersect the tree. Though performing an edge-slide is a constant-time operation, checking its validity might be not. Also we can so far only give an exponential upper bound for the complexity of finding the shortest sequence of slides between two trees.

Acknowledgments

We would like to thank Ferran Hurtado for his presentation 'Flip' as the Paul

Erdös lecture at CCCG 2003 in Halifax, Canada. This inspired us to (continue) work on the presented topic. We are grateful to Franz Aurenhammer and Hannes Krasser for enjoyable discussions and carefully reading the manuscript.

References

[1] O.Aichholzer, F.Aurenhammer, F.Hurtado *Sequences of spanning trees and a fixed tree theorem.* Computational Geometry: Theory and Applications, 21(1-2):3–20, 2002.

[2] D.Avis, K.Fukuda *Reverse search for enumeration.* Discrete Applied Mathematics 65: 618-632, 1996.

[3] O.Aichholzer, K.Reinhardt *A quadratic distance bound on sliding between crossing-free spanning trees - extended abstract.* In Proc. 20th European Workshop on Computational Geometry EWCG '04: 13-16, Sevilla, Spain, 2004.

[4] W.Goddard, H.C.Swart *Distances between graphs under edge operations.* Discrete Mathematics 161: 121–132, 1996.

[5] S.Hanke, T.Ottmann, S.Schuierer *The edge-flipping distance of triangulations.* Journal of Universal Computer Science 2 (1996), 570-579.

[6] M.C.Hernando, M.E.Houle, F.Hurtado *On local transformation of polygons with visibility properties.* Proc. 6th International Computing and Combinatorics Conference COCOON'00, Springer LNCS 1858: 54–63, 2000.