Happy Reconfiguration

GADA Lab, University of Manitoba

August 2024

1 Preliminaries and work so far:

Let P be a set of n points in convex position, i.e., the convex hull of P is an n-sided polygon (or n-gon). We will represent the convex hull of P by $CH(P)$.

Definition 1.1 (Caterpillar)**.** *A tree T is called a* caterpillar *if removing its leaves (and their incident edges) produces a path, which is called the* spine *of T.*

Definition 1.2 (Interior Edge)**.** *An edge e which is not on the convex hull of a graph drawing in a convex point set P is called an **interior edge**.*

Definition 1.3 (Boundary Edge)**.** *An edge e' which is not an interior edge is called a **boundary edge**.*

Lemma 1.4. *Every non-crossing spanning tree of a set of points in convex position has at least two boundary edges.*

Definition 1.5. *An interior edge e cuts $CH(P)$ into two convex regions $R_L(e)$ and $R_R(e)$ on convex point sets $P_L(e)$ and $P_R(e)$, respectively. Given a non-crossing spanning tree T of P, the* restriction *of T to $P_L(e)$ (resp., $P_R(e)$) is the subgraph of T induced by the vertices in $P_L(e)$ (resp., $P_R(e)$).*

Observation 1.6. *The restriction of T to $P_L(e)$ (resp., $P_R(e)$) is a non-crossing spanning tree of $P_L(e)$ (resp., $P_R(e)$).*

Lemma 1.7. *Given a non-crossing spanning tree T of P, any interior edge e cuts the n-gon into exactly two convex polygons who share e as a common edge and each such polygon has at least two boundary edges, one of which is e.*

Proof for 1.7.
Consider the two regions $R_L(e)$ and $R_R(e)$ that e cuts the $CH(P)$ into. By Lemma 1.4, the restriction of T to $P_L(e)$ has at least two boundary edges, one of which is e. Thus, the restriction of T of $P_L(e)$ has at least one boundary edge of $CH(P)$. A similar argument holds for the restriction of T to $P_R(e)$. □

Theorem 1.8 (Graph drawing of caterpillar)**.** *Given a convex point set P, if a graph drawing of a tree T has exactly 2 edges on the convex hull, then T is a caterpillar. In addition, every caterpillar graph can be redrawn such that exactly 2 edges remain on the convex hull.*

Proof for 1.8.

We will first show that if a graph drawing of a non-crossing spanning tree T has exactly 2 edges on the convex hull, then T is a caterpillar.

Add proper notation for restriction

Let the two boundary edges of a graph drawing of a non-crossing spanning tree T be $b_1 = (u_1, v_1)$ and $b_2 = (u_2, v_2)$. We will consider the cases of when b_1 and b_2 are adjacent and when they are not:

Case 1: b_1 and b_2 are adjacent.
Note that if b_1 and b_2 are adjacent, then b_1 and b_2 share a common vertex, say u_1. If b_1 and b_2 are adjacent, then the graph drawing of a tree T is a star. We can show this statement holds true by considering an interior edge e that does not contain u_1, then we will obtain that both b_1 and b_2 are only on one of the polygons obtained from when we cut the convex n-gon obtained from the convex point set P using e. This would imply that one of the polygons will have 3 boundary edges: b_1, b_2, and e which is a contradiction from Lemma 1.7 and hence, we obtain a star. Since a star is a caterpillar, we are done.

Case 2: b_1 and b_2 are not adjacent.
Let $P = (u_1, v_1, \ldots, v_2, u_2)$ be the longest path starting from the endpoint of b_1 and ending at the endpoint of b_2. Consider the polygons obtained from cutting the n-gon using P. We note that each polygon contains exactly two boundary edges (both of which are from the path) and these two edges are adjacent. Then, from Case 1, we see that these edges of the polygons form a star. Since we have stars chained together by a path, once we remove all the leaves of each star, we simply get P. Hence, we also get a caterpillar in this case.

Since we have a caterpillar in both cases, this shows that if a graph drawing of a non-crossing spanning tree T has exactly 2 edges on the convex hull, then T is a caterpillar.

Now, we show that every caterpillar graph can be drawn such that exactly 2 edges remain on the convex hull.

Let $P = \{p_1, p_2, \ldots, p_n\}$ be the spine of caterpillar graph. Let L_i be the set of all leaves of the vertex p_i. Let B be the 4-gon formed by setting p_1, p_2, p_{n-1}, and p_n as its corners, let $CH(B)$ represent the convex hull of B.

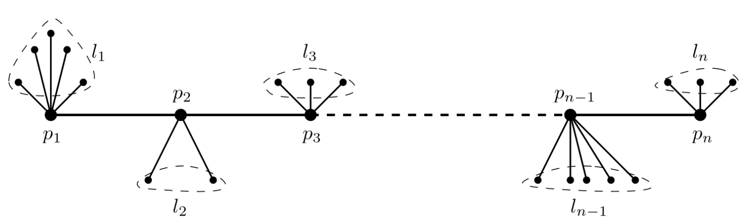

Figure 1: A caterpillar with spines and leaves labelled.

Now, align P such that $b_1 = (p_1, p_2)$ and $b_2 = (p_{n-1}, p_n)$. WLOG, assume that n is even and partition P into two subsets P_E and P_O such that all p_i of P where i is even are in P_E and all p_j of P where j is odd are in P_O. In addition, set all the elements of L_i to be in P_E if i is odd and set them to be in P_O if i is even. Note that p_1 and p_{n-1} are both in P_O and p_2 and p_n are both in P_E. Now, align all the elements of P_E from P such that they lie in the line joining p_1 and p_{n-1} from $CH(B)$. Similarly, align all the elements of P_O from P such that they lie in the line joining p_2 and p_n from $CH(B)$. Then, align all the elements of P_E that are leaves such that they fall in between p_i and p_{i+1} in the line. Do so similarly for P_O.

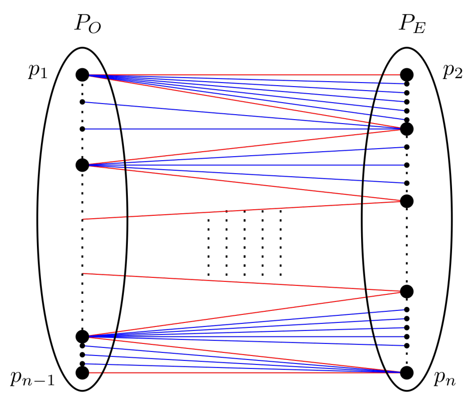

Figure 2: Partitioned caterpillar. The blue edges are the edges with leaves. The red edges are the spine.

Note that P_E and P_O are disjoint, independent, and non-crossing. Also, observe that there are exactly 2 boundary edges in B, namely b_1 and b_2. And so, we are done. $\square$

2 Happy Edge Conjecture for Convex Point Sets

An edge e is *happy* if e lies in $T_I \cap T_F$.

Lemma 2.1. *For any point set P in convex position and any two non-crossing spanning trees T_I and T_F of P, there is a minimum flip sequence from T_I to T_F such that no happy edge is flipped during the sequence.*

Aichholzer et al. [1] conjectured that the statement in Lemma 2.1 holds and proved it for the case of happy edges on the convex hull. We prove that their conjecture indeed holds in the general case.

Proof.
We prove by induction on the number of *interior* happy edges, that is, those happy edges that are not on the convex hull. For the base case, if there are no

Add proper notation for restriction

interior happy edges, then we use the result of Aichholzer et al. [1] that there is a minimum flip sequence from T_I to T_F without flipping the happy edges.

Assume now that there are k interior happy edges. Select one such edge e (assume wlog. that it is vertical) and observe that no other edge of T_I or T_F crosses e. Therefore, we can divide the point set P into P_l and P_r of all the points in P that are to the left and to the right of e (including the endpoints of e) respectively. Similarly, we can divide the tree T_I into subtrees $T_{I,l}$ and $T_{I,r}$ as well as T_F into non-crossing spanning subtrees $T_{F,l}$ and $T_{F,r}$ embedded onto P_l and P_r, respectively (Observation 1.6). Notice that $T_{I,l}$ and $T_{F,l}$ as well as $T_{I,r}$ and $T_{F,r}$ have one less happy edge in their interiors, therefore, by induction there is a minimum flip sequence from $T_{I,l}$ to $T_{F,l}$ on P_l and from $T_{I,r}$ to $T_{F,r}$ on P_r that do not flip their happy edges. Observe now that any minimum flip sequence from T_I to T_F can be partitioned into flips only on P_l and only on P_r since no edge can cross e. Therefore, we can combine the minimum flip sequence from $T_{I,l}$ to $T_{F,l}$ on P_l and from $T_{I,r}$ to $T_{F,r}$ on P_r to obtain a minimum flip sequence from T_I to T_F. □

3 Greedy algorithm [DEPRECATED]

Rough draft of the algorithm as discussed:

1. Select lowest intersecting blue edge and make the number of intersections to the blue edge 0 by flipping all red edges intersection with it and place it on the boundary. (Idea: While flipping all intersecting red edges, conserve the endpoint closest to the boundary while trying to make it connected).

2. Now, take the red edge with the most number of intersections and swap it with the blue edge with 0 intersections. Prioritise the edges placed in the boundary.

   - If this step creates a cycle then remove the red edges on the boundary and *perfect* flip them into the sub-problem on the region formed by the last removal of the blue edge.

4 Algorithm for converting any caterpillar into any other non-crossing spanning tree in $\frac{3n}{2} + c$ steps

Definition 4.1. *If R is a region, then the length of the path formed on $CH(R)$ is denoted by $\partial(R)$.*

Definition 4.2 (Eccentricity of reconfiguration graphs)**.** *The total distance (i.e, the number of edge flips) from an initial non-crossing spanning tree T_I to a final non-crossing spanning tree T_F is defined to be the **eccentricity** of T_I.*

Furthermore, the eccentricity of a graph with i boundary edges is denoted by e_i, for $i \in \mathbb{N}$.

Proposition 4.3 (Bound of eccentricity of caterpillar graphs). *The eccentricity of a non-crossing drawing of any caterpillar graph to any other non-crossing spanning tree is bounded by* $\frac{3n}{2} \pm c$*, for* $c \in O(1)$*.*

Proof. General idea:

1. Consider the 2 boundary edges of T_I, call them b_1 and b_2. Also, note that we shall color all the edges of T_I as red and color all the edges of T_F as blue.

2. Notice that b_1 and b_2 divides $CH(T_I)$ into 2 regions, call them R_1 and R_2.

3. Consider the minimum of $\partial(R_1)$ and $\partial(R_2)$. WLOG, say that $\partial(R_1) \leq \partial(R_2)$. Note that $\partial(R_1) \leq \partial(R_2) \leq \dfrac{n}{2}$, where n is the number of vertices. Now, consider R_1:

4. "Shifting the spine." [Iterative argument] (Maximum number of steps taken is at most $\frac{n}{2}$)

   - Add a red edge on $CH(R_1)$. Since T_I is a tree, adding an edge creates a cycle. To resolve the cycle, delete an edge of that cycle that is not a boundary edge. Note that the resulting tree is still a connected, non-crossing, spanning tree.
   - Keep repeating the previous step until $CH(R)$ has its boundary completely covered by red edges.
   - The end result will be a caterpillar graph with it's spine on $CH(R_1)$ and all the leaves forming an edge with the spine on $CH(R_1)$.

5. We now claim that every edge flip that occurs from now on is a (pseudo) perfect flip. [It is pseudo in the sense that it isn't flipping to the actual T_F but instead converging from both directions to a single tree, say T'_F, in n flips.]

6. "Shifting" T_F [Iterative + Inductive argument]

   - Now, consider T_F, we shall apply a similar iterative argument as before.
   - Add a blue edge on $CH(R)$. Similarly to the previous case, we will obtain a cycle. To resolve the cycle, delete a non-boundary edge of the cycle. Note that the resulting graph is a connected non-crossing spanning tree.
   - Keep repeating the previous step until $CH(R)$ has its boundary completely covered by blue edges.
   - Note that every single edge that we flipped only had to be flipped once so each edge only contributes 1 flip.

7. Set the cardinal directions as b_1 as the 'left', b_2 as the 'right', R_1 as the 'top', and R_2 as the 'bottom'.
   - From 4, we know that there exists at least 1 red edge that has 0 crossings. From the algorithm outlined in 4, pick the blue edge that is connected to the endpoint of the red edge that has 0 crossings and flip the blue edge to the red edge.
   - We now have two smaller subproblems towards the region to the left of the flipped blue edge and towards the right of the flipped blue edge.
   - Note that those two sub regions have the same criteria as our main graph and thus we can keep applying 4 and flipping blue edges to red edges as shown above.
   - Also, note that each flip is a perfect flip so each edge only contributes 1 flip.
8. Finally, from the above two steps, we know that we can flip the blue tree into the red shifted spine tree in at most n steps. Combining everything, we get that the total number of flips from any red non-crossing spanning caterpillar to any blue non-crossing spanning tree is at most $\frac{n}{2} + n = \frac{3n}{2}$ flips.

$\square$

Lemma 4.4. *After the shifting the spine step, there will always be at least 1 red edge that has 0 crossings.*

Proof. All the variables follow from the shifting the spine step. Set the cardinal directions as b_1 as the 'left', b_2 as the 'right', R_1 as the 'top', and R_2 as the 'bottom'.

- Pick the closest vertex from b_1 and R_2, call it u_1.
- Note that $u_1 \notin R_1$. However, from our "Shifting the spine" step, we know that it is connected to the spine formed in R_1 by an edge, call it e_1. Now, follow along from u_1 to e_1 and note each time there is a crossing from a blue edge to e_1. Since we are dealing with at most $\frac{n}{2}$ remaining blue edges, there will be at most $\frac{n}{2}$ crossings which is a finite amount. Now, take the last crossing encountered and follow the blue edge that caused the crossing all the way to the endpoint of the blue edge towards b_1.
- Repeat the above step until you find a red edge that has 0 crossings.
- Say, for contradiction, that after repeating the above steps we do not find a red edge that has 0 crossings. This implies that we followed a blue edge to an endpoint in b_2. Hover since b_2 already had both red and blue edges, this implies that there exists a cycle in the blue tree which is a contradiction.

$\square$

References

[1] Oswin Aichholzer, Brad Ballinger, Therese C. Biedl, Mirela Damian, Erik D. Demaine, Matias Korman, Anna Lubiw, J. Lynch, Josef Tkadlec, and Yushi Uno. Reconfiguration of non-crossing spanning trees. *ArXiv*, abs/2206.03879, 2022.