On Straightening Low-Diameter Unit Trees*

Sheung-Hung Poon

Department of Mathematics and Computer Science, TU Eindhoven,
5600 MB, Eindhoven, the Netherlands.
spoon@win.tue.nl

A *polygonal chain* is a sequence of consecutively joined edges embedded in space. A *k-chain* is a chain of k edges. A *polygonal tree* is a set of edges joined into a tree structure embedded in space. A *unit tree* is a tree with only edges of unit length. A chain or a tree is *simple* if non-adjacent edges do not intersect.

We consider the problem about the reconfiguration of a simple chain or tree through a series of continuous motions such that the lengths of all tree edges are preserved and no edge crossings are allowed. A chain or tree can be *straightened* if all its edges can be aligned along a common straight line such that each edge points "away" from a designed leaf node. Otherwise it is called *locked*. Graph reconfiguration problems have wide applications in contexts including robotics, molecular conformation, rigidity and knot theory. The motivation for us to study unit trees is that for instance, the bonding-lengths in molecules are often similar, as are the segments of robot arms.

A chain in 2D can always be straightened [4, 5]. In 4D or higher, a tree can always be straightened [3]. There exist trees [2] in 2D and 5-chains in 3D that can lock. Alt et al. [1] showed that deciding the reconfigurability for trees in 2D and for chains in 3D is PSPACE-complete. However the problem of deciding straightenability for trees in 2D and for chains in 3D remains open.

It is easy to verify that a tree of diameter at most 3 in 2D or 3D can always be straightened. In this paper, we show that some tree of diameter 4 in 2D or 3D can lock, and a unit tree of diameter 4 in 2D can always be straightened.

In 2D, even a tree with diameter as low as 6 can lock [2] as shown in Figure 1 (a). We present a locked tree of diameter 4 in Figure 1 (b), which simulates the tree in (a). It can be shown locked using the same technique as the proof for (a) by assigning the corresponding equilibrium stresses to the tree edges. In 3D, a 5-chain can lock [2]. We present a 3D locked tree of diameter 4, which is shown in Figure 1 (c).

We now consider the straightenability of a unit tree T of diameter 4 in 2D. The *center* of tree T, denoted by o, is the middle vertex of any 4-chain in T. We call a path connecting the center to a leaf a *branch* of T. A *direct straightening* of branch $B = ouv$ in T means to rotate v around u until ouv is straightened by passing through the smaller angle. We denote the sweeping region for directly straightening B by $S(B)$. The direct straightening of B is *interfered* by another branch B' if $S(B) \cap B' \neq \emptyset$. There are two kinds of interferences depending on whether B and B' are of the same turn. We say that B' *follows* (resp. *covers*) B

* This research was supported in part by the Netherlands' Organisation for Scientific Research (NWO) under project no. 612-065-307.

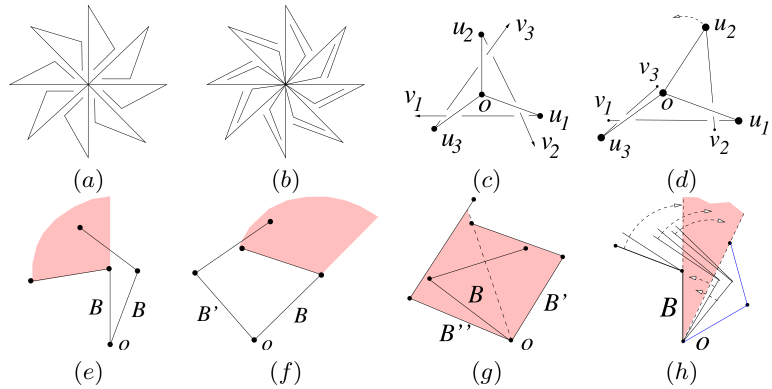

Fig. 1. $(a), (b)$ 2D locked trees. (c) A 3D locked tree of diameter 4, where $d(o, u_i) = 1$ & $d(u_i, v_i) \geq 2$. (d) Straightening unit-tree version of (c). (e) B' follows B. (f) B' covers B. (g) B'' covers B' & B' covers B. (h) Straightening uncovered branch B and all its following branches.

if B is interfered by B' of the same (resp. opposite) turn. See Figure 1 $(e), (f)$ for illustration.

Our algorithm to straighten T relies heavily on the observation of a nice nesting structure on covering relation. Suppose B'' covers B' which in turn covers B. Then B is nested inside the area enclosed by B and B'', which is the shaded area as shown in Figure 1 (g). Therefore the last branch in a maximal covering sequence is always uncovered. Our algorithm proceeds by successively straightening an uncovered branch and all its following branches. The procedure to straighten a uncovered branch is shown in Figure 1 (h). The whole algorithm can be designed to run in $O(n)$ moves and $O(n \log n)$ time, where n is the number of tree edges.

In 3D, we conjecture that a unit tree of diameter 4 can always be straightened. In particular, it is not hard to see that the unit-tree version of Figure 1 (c) can be straightened. We first rotate v_1 around u_1 until $u_1 v_1$ is very close to ou_3, and then rotate v_3 around u_3 until v_3 is very close to o. Consequently we can rotate u_2 around o to draw ou_2v_2 out. We further conjecture that a unit tree of any diameter in 2D or 3D can always be straightened.

References

1. H. Alt, C. Knauer, G. Rote, and S. Whitesides. The Complexity of (Un)folding. *Proc. 19th ACM Symp. on Comput. Geom. (SOCG)*, 164–170, 2003.
2. T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. A Note on Reconfiguring Tree Linkages: Trees can Lock. *Disc. Appl. Math.*, 117:1-3, 293–297, 2002.
3. R. Cocan and J. O'Rourke. Polygonal Chains Cannot Lock in 4D. *Comput. Geom.: Theory & Appl.*, 20, 105–129, 2001.
4. R. Connelly, E.D. Demaine, and G. Rote. Straightening Polygonal Arcs and Convexifying Polygonal Cycles. *Disc. & Comput. Geom.*, 30:2, 205–239, 2003.
5. I. Streinu. A combinatorial approach for planar non-colliding robot arm motion planning. *Proc. 41st ACM Symp. on Found. of Comp. Sci. (FOCS)*, 443–453, 2000.