An Energy-Driven Approach to Linkage Unfolding

Jason H. Cantarella* Erik D. Demaine† Hayley N. Iben‡ James F. O'Brien§

1 Linkage Reconfiguration. Consider a planar linkage of rigid bars joined at flexible joints to form a collection of tangled but noncrossing arcs and cycles (polygonal chains). The linkage is free to move in any way that preserves the bar lengths and causes no two bars to cross. The Carpenter's Rule Theorem [3, 4] says that such a linkage can be continuously folded into every possible configuration, or equivalently, that any configuration can be folded to straighten the arcs and convexify the polygons.

The algorithmic side of this theorem remains relatively open. The original proof of [3] is algorithmic but requires solving an ordinary differential equation defined by a convex optimization. This motion has the advantage of being "canonical", in particular preserving any symmetries present in the original linkage and expanding all distances between pairs of vertices. The alternative approach of [4] leads to a more-efficient algorithm, involving the computation of polynomially many algebraic motions of degree $\Theta(n)$. On the other hand, this motion does not preserve symmetries in the linkage. Neither of these motions can ever be computed explicitly because of their inherent complexity; rather, a desired number of snapshots along the motion can be computed to desired accuracy, and the precise behavior between these snapshots remains opaque.

2 Our Results. In this paper, we introduce a new energy-driven approach to straightening arcs and convexifying cycles that establishes stronger mathematical, algorithmic, and practical results.

On the mathematical side, we obtain a completely smooth (C^∞) motion. In contrast, the motions of [3] are piecewise-C^1 and the motions of [4] are piecewise-C^∞, with polynomially many pieces; so neither is even C^1. In this sense, the motions produced by our energy-based approach are particularly natural and canonical, also preserving symmetries in the linkage, and proving contractibility of the configuration space in many cases. The motions are not expansive, but this relaxation seems key to achieving our results.

On the algorithmic side, we obtain the first algorithm that constructs an explicit motion. This motion is piecewise-linear (the simplest possible type of motion) and approximates the smooth motion arbitrarily closely. Furthermore, the motion can be computed exactly on a machine supporting real arithmetic, in contrast to all previous approaches. Each piece of the motion is computed in $O(n^2)$ time, while the number of pieces in the motion depends on the geometric complexity of the instance in addition to n. This geometric dependence is necessary for any explicit construction of a motion. In comparison, the approach of [3] depends on similar parameters, but the running time is significantly larger. The approach of [4] only depends on n, plus the desired "snapshot rate" of the motion, but the running time is exponential in n (from the manipulation of algebraic curves of degree $\Theta(n)$).

On the practical side, our algorithm is easy to implement, involving a straightforward computation of the gradient of an energy function. We have implemented the algorithm as a Java applet [1]. In addition, the algorithm corresponds to a natural physical process, in which vertices repel bars (and vice versa) as if they all were objects with similar electrostatic charges, and the system evolves in the usual model of physics.

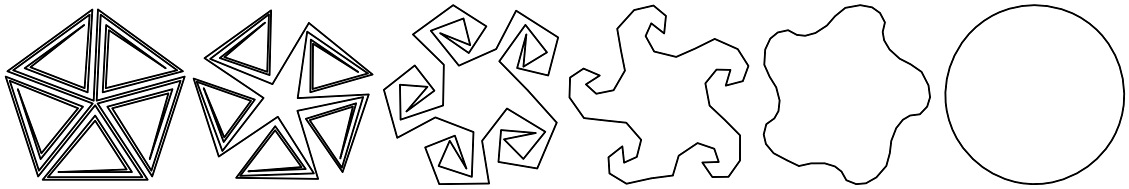
3 Overview. The basic idea of our approach is to define an *energy function* on the configurations of the linkage, satisfying three properties:

1. the energy is infinite when the linkage crosses itself;
2. the energy is minimum when the linkage is in the desired configuration (straight or convex); and
3. expansive motions decrease energy.

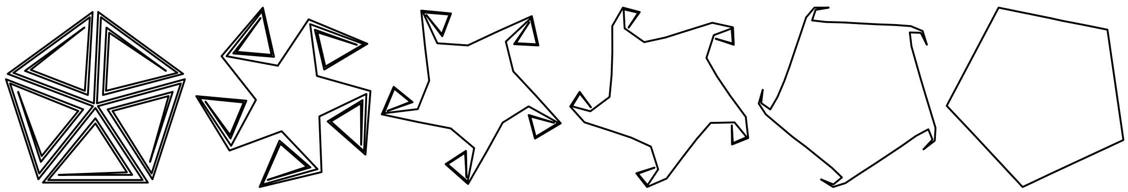
The third property, together with the existence of expansive motions [3], establishes the existence of motions that decrease energy. We apply gradient descent to follow the motion that decreases energy most steeply. The first property implies that this energy-decreasing motion will avoid self-intersection, and the second property implies that we eventually reach the desired configuration. The formalization of this argument requires much more detail and care; refer to the full paper.

4 Experiments. We compared an implementation of [3] based on the CPLEX barrier solver for quadratic programs to a C++ implementation of our energy approach, on two examples of closed chains. One example of the resulting animations is shown in

*University of Georgia, *cantarel@math.uga.edu*
†Massachusetts Institute of Technology, *edemaine@mit.edu*
‡University of California, Berkeley, *iben@eecs.berkeley.edu*
§University of California, Berkeley, *job@eecs.berkeley.edu*

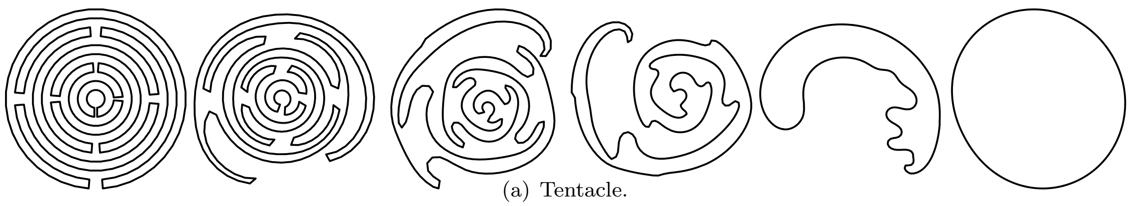


(a) Doubled tree with energy method.

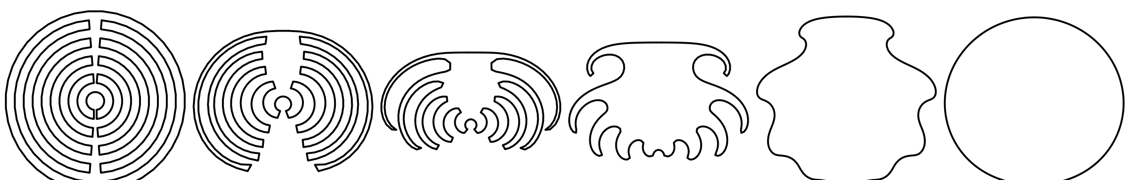


(b) Doubled tree with CDR method.

Figure 1: A comparison of convexification by our method and by CDR. To maximize visibility, the animation zooms as time proceeds; in fact, all edge lengths remain constant. This example has 50 vertices.



(a) Tentacle.



(b) Spider.

Figure 2: Other examples of straightening and convexification computed with our method. To maximize visibility, the animation zooms as time proceeds; in fact, all edge lengths remain constant. These examples have 380 vertices.

Figure 1. We omit details of the running time here. A short summary is that the energy method is much faster, in each step requiring only $O(n^2)$ time instead of solving a quadratic program, as well as more accurate, avoiding the approximation inherent in the solution to the differential equation. To illustrate the scalability of the energy approach, we show some additional examples in Figure 2.

5 Conclusion. We have presented a simpler, more efficient, and practical method to unfold linkages made up of arcs and cycles. While the motion is not globally expansive, its minimization of energy attempts to balance distances and reconfigure the linkage more “organically.” In the formal sense, this motion is the first that is completely smooth (C^∞) in the limit.

One interesting question about our motion is to determine the shape of the final minimum-energy configuration of a cycle. In contrast to [3] or [4], which have unpredictable final configurations, we might expect that our energy method results in a cycle that best approximates a regular polygon, that is, causes the joints to lie on a common circle. See [2] for other results along these lines. From our experiments, this expectation seems false, but a combination of our energy function with a term involving the area of the polygon may lead to such a result.

References.

[1] `http://www.cs.berkeley.edu/~job/Projects/Unfold/`.

[2] Aaron Abrams, Jason Cantarella, Joseph H.G. Fu, Mohammad Ghomi, and Ralph Howard. Circles minimize most knot energies. *Topology*, 2002. To appear.

[3] Robert Connelly, Erik D. Demaine, and Günter Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete & Computational Geometry*. To appear. Preliminary version appeared at FOCS 2000.

[4] Ileana Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 443–453, Redondo Beach, California, November 2000.