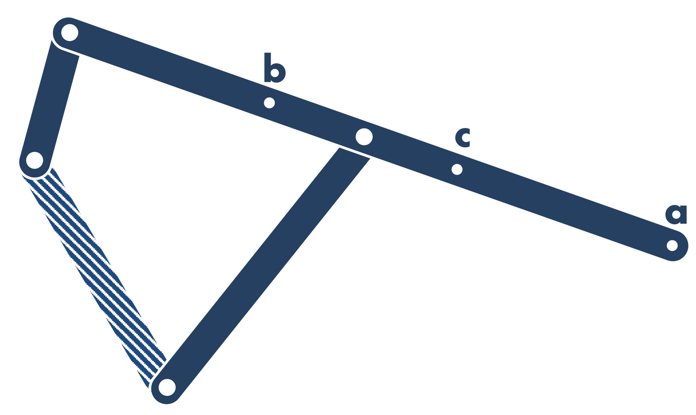
Four-Bar Linkage Synthesis Using Non-Convex Optimization

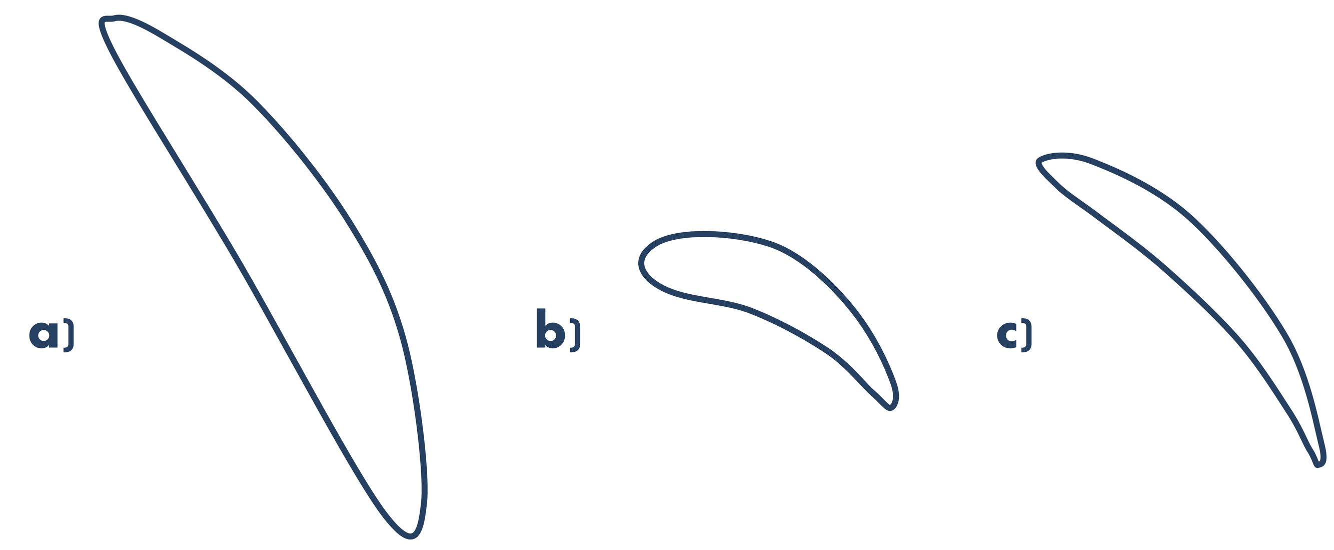
Vincent Goulet[1] Wei Li[2] Hyunmin Cheong[2] Francesco Iorio[2] Claude-Guy Quimper[1]

1 Université Laval
2 Autodesk Research

The problem

Four-bar linkages are simple mechanical systems able to output a large variety of shapes, called coupler curves. The path synthesis problem takes a curve as input from the user and finds the best matching linkage.



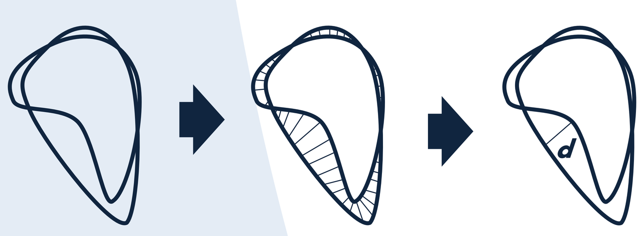


The solution

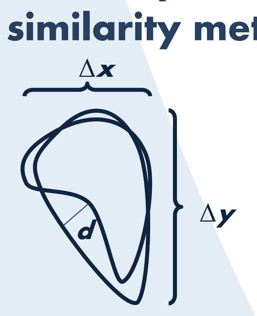
Curve Similarity Metric

A metric is needed to compare the input curve to the coupler curve of the output four-bar linkage. We use the Hausdorff distance and normalize it to make it independent of scale.

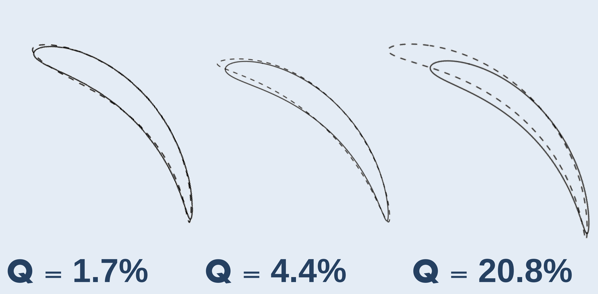
Calculation of the Hausdorff distance: For all points on the input curve, find the shortest distance to the coupler curve. The Hausdorff distance is the largest of these distances.



Since d is dependent on scale, we divide it by the largest dimension of the input curve to obtain our similarity metric Q:



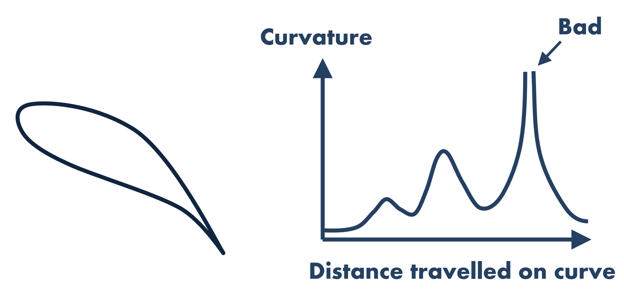
The lower the Q value, the better the match. A perfect match has Q = 0%. In this work, curves with Q lower than 5% are considered a satisfactory match.



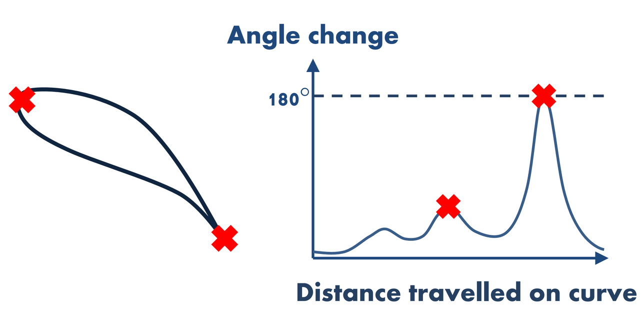
Curve Sampling Strategy

The user-input curve is reduced to an array of carefully chosen points. Choosing the right points reduces the size of the model while sacrificing as little accuracy as possible.

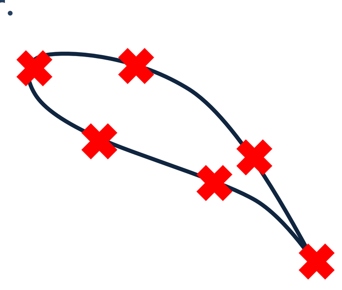
We find that points at maxima of curvature on the curve have the most impact on the Q metric.



Curvature gets inconveniently large at sharp turns. To solve this problem, we approximate the curve with small segments and calculate the difference of angle of adjacent segments. The maxima are then identified and filtered

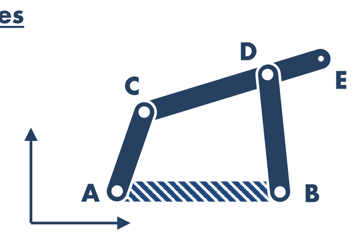


Finally, some points are evenly spread between the identified features to depict general behavior.



Model

Variables



The linkage is fully determined by the coordinates of the fixed pivots A and B and the lengths of the links:

$$A_x \,;\, A_y \,;\, B_x \,;\, B_y$$
$$AB \,;\, AC \,;\, BD \,;\, CD \,;\, CE$$

For each sample point, the linkage assumes a different position where points C, D and E have different coordinates:

$$C_{xi} \,;\, C_{yi} \,;\, D_{xi} \,;\, D_{yi} \,;\, E_{xi} \,;\, E_{yi}$$

An error variable is defined as the upper limit to the distance point E gets to any sample point: e

Constraints

The links are encoded by setting the distance between coordinates equal to the corresponding length.

$$(A_x - B_x)^2 + (A_y - B_y)^2 = AB^2$$
$$(A_x - C_{xi})^2 + (A_y - C_{yi})^2 = AC^2$$
$$(B_x - D_{xi})^2 + (B_y - D_{yi})^2 = BD^2$$
$$(C_{xi} - D_{xi})^2 + (C_{yi} - D_{yi})^2 = CD^2$$

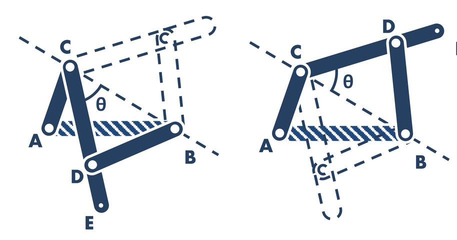
Since E is collinear with C and D, vectors from C to D and from C to E have proportional components:

$$CD \cdot (E_{xi} - C_{xi}) = CE \cdot (D_{xi} - C_{xi})$$
$$CD \cdot (E_{yi} - C_{yi}) = CE \cdot (D_{yi} - C_{yi})$$

The error is related as such to the distance from E_i to the corresponding sample point, noted T_i :

$$(T_{xi} - E_{xi})^2 + (T_{yi} - E_{yi})^2 < e$$

The positions found for each sample point might belong to either of two possible ways to assemble the linkage. This is known as *branch defect*.



For the first assembly, sin(θ) is always negative, and for the second it is always positive. The sine is related to the cross-product of vectors.

$$\overrightarrow{CB} \times \overrightarrow{CE} > 0$$

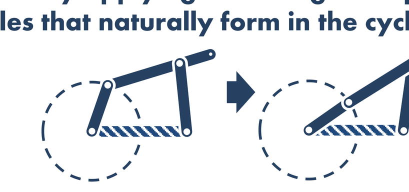
Expressed with the variables of the model, this yields:

$$(T_{xi} - C_{xi})\,(B_{yi} - C_{yi}) > (T_{yi} - C_{yi})\,(B_{xi} - C_{xi})$$

The Grashof condition states that the shortest link of a four-bar linkage is able to fully rotate if its combined length with the longest link is shorter than the combined length of the two remaining links. To have AC rotate fully, we need:

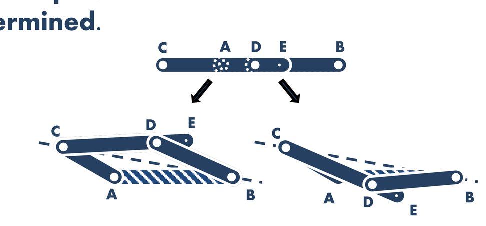
$$AC < AB \qquad AC < BD \qquad AC < CD$$

The other constraints needed for the condition are obtained by applying the triangle inequality to triangles that naturally form in the cycle:



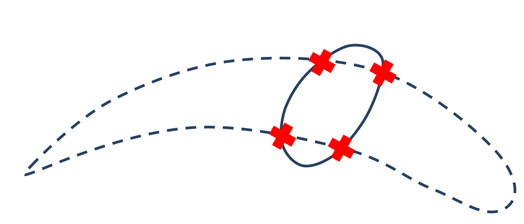
$$CD + BD > AC + AB + s$$
$$AB + CD > AC + BD + s$$
$$AB + BD > AC + CD + s$$

A small security constant s is added so that the lengths at the right are always at least a bit smaller than those at the left. Otherwise, the mechanism could collapse, in which case its behavior is undetermined.



Area Constraint

If the number of sample points is small, the dashed coupler curve should not be a valid solution, since its Q value is very high.



Adding more points would solve this problem, but would increase the computation time.

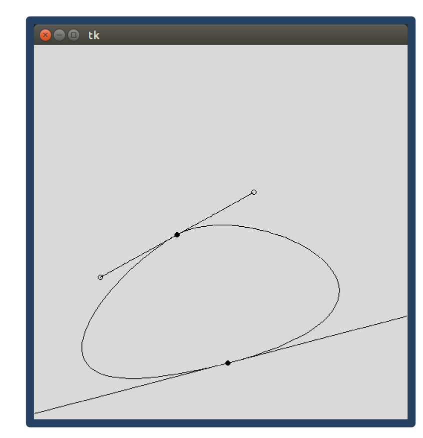
Constraining the area of the curve would also be helpful on that regard. The following expression for the area of a coupler curve was discovered and proved:

$$Area = \pi \cdot AC^2 \cdot \left(\frac{CE}{CD} - 1\right)$$

This simple redundant constraint dramatically reduces the search space and has low complexity. This is a novel way to analytically relate the coupler curve to the four-bar linkage.

Software

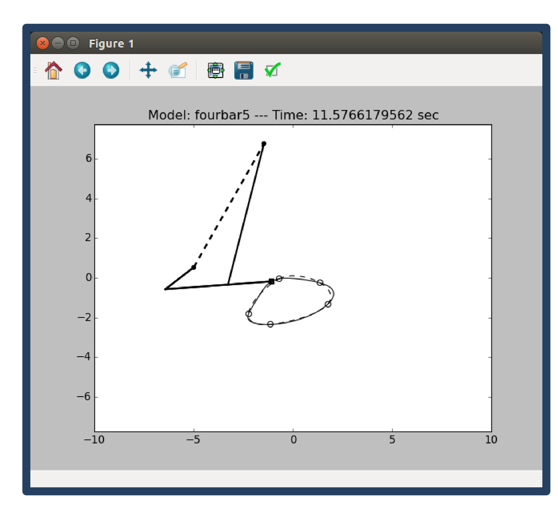
A simple Python software implementing the solution was developed. The user is prompted to draw a Bezier curve.

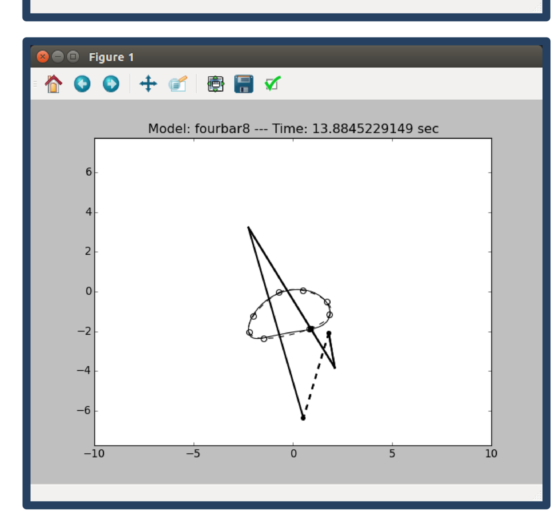


The software launches several sampling options at once.

```
Launching process 0: model 'fourlin' with 4 points
Launching process 1: model 'fourlin' with 5 points
Launching process 2: model 'fourlin' with 8 points
Process 0 completed after 2.312 seconds with score 7.7 %
Process 1 completed after 12.494 seconds with score 4.3 %
Process 2 completed after 14.724 seconds with score 3.0 %
Solving done.
```

The Q metric is calculated for the returned solutions. If it is below the acceptance threshold, the execution is stopped and the solution is returned as an animation.





Experimentation

Choice of solver

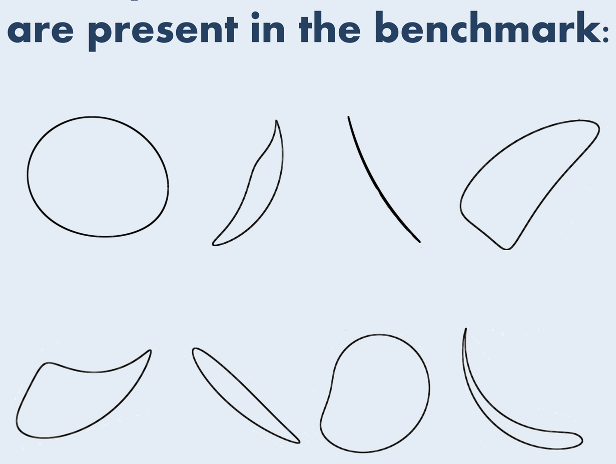
The model has real variables and non-linear, non-convex constraints. Compatible solvers include:

- Ibex
- RealPaver
- alphaBB
- SCIP
- BARON
- Couenne
- LindoAPI

Couenne showed the best performance. Its strategy features bound tightening, linearization and branch and bound.

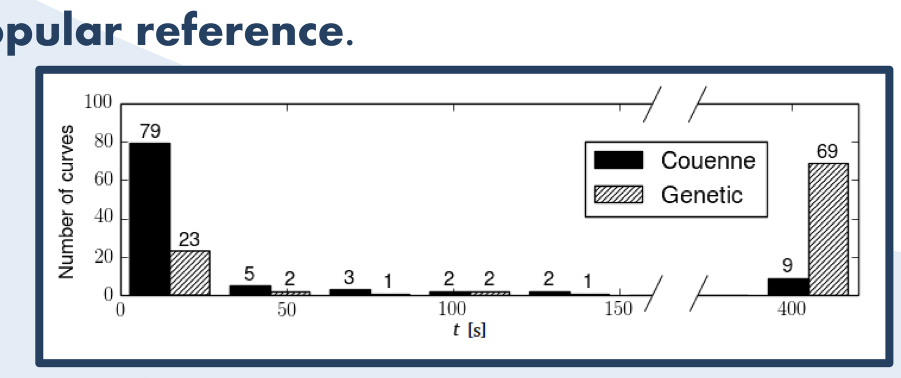
Benchmark

The benchmark is composed of 100 randomly generated coupler curves such that all types of reproducible shapes are represented. These curves are present in the benchmark:



Comparison with evolutionary algorithm

Out of many alternative approaches from related works, the evolutionary algorithm presented by Cabrera et al. appears as a popular reference.



Our results show that the non-convex optimization solved much more curves. Only 9 curves timed out at 400 s with *Couenne*, while 69 timed out using the evolutionary algorithm.

Area constraint evaluation

Sampling	Area	$Q<5$ % 5 s	60 s	400 s	No solution
{4, 5, 6}	**Yes**	**59**	**83**	**92**	**0**
{4, 5, 6}	No	37	58	63	0
{6, 7, 8}	Yes	51	81	89	1
{6, 7, 8}	No	50	68	78	1
{10, 12, 16}	Yes	30	59	69	11
{10, 12, 16}	No	33	57	66	14

The results show that the constraint had the most impact when there were fewer sample points. The best combination was using few points and the constraint.