Computing Animations of Linkages with Rotational Symmetry

Sean Dewar
Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Linz, Austria
sean.dewar@ricam.oeaw.ac.at

Georg Grasegger
Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Linz, Austria
georg.grasegger@ricam.oeaw.ac.at

Jan Legerský
Johannes Kepler University Linz, Research Institute for Symbolic Computation (RISC), Linz, Austria
Department of Applied Mathematics, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic
jan.legersky@risc.jku.at

Abstract

We present a piece of software for computing animations of linkages with rotational symmetry in the plane. We construct these linkages from an algorithm that utilises a special type of edge colouring to embed graphs with rotational symmetry.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Flexibility, Linkages, Symmetry

Digital Object Identifier 10.4230/LIPIcs.SoCG.2020.77

Category Media Exposition

Related Version A proof of the theorem can be found in [1], `https://arxiv.org/abs/2003.09328`.

Supplementary Material An implementation can be found in [3, 4], `https://doi.org/10.5281/zenodo.3719345`.

Funding *Sean Dewar*: Supported by the Austrian Science Fund (FWF): P31888.
Georg Grasegger: Supported by the Austrian Science Fund (FWF): P31888 and W1214-N15.
Jan Legerský: Supported by the Austrian Science Fund (FWF): P31061 and by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675789.

1 Introduction

A *framework* is a pair (G, p) where G is a (finite simple) graph and $p : V(G) \to \mathbb{R}^2$ – the *placement* of G – is a map where $p(u) \neq p(v)$ if uv is an edge. A framework is a *linkage* if there exists a continuous motion of its placed vertices that preserves the distances between each pair of vertices that share an edge, and the motion is not a rigid body motion of the framework; if such a motion does not exist then the framework is *rigid*. It was shown in [6] that a framework (G, p) with a *generic* placement of vertices (i.e. the set of coordinates of p is an algebraically independent set over the rational numbers) is rigid if and only if G contains a Laman graph as a spanning subgraph. This does not inform us whether we can construct a linkage from a graph; for example, any generic placement of the complete bipartite graph $K_{4,4}$ is rigid, however we can construct linkages from $K_{4,4}$ [2].

36th International Symposium on Computational Geometry (SoCG 2020).
Editors: Sergio Cabello and Danny Z. Chen; Article No. 77; pp. 77:1–77:4
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

To determine whether a graph can be the graph of a linkage we introduce a special class of edge colourings. A red-blue colouring of the edges of a graph is a *NAC-colouring* if each colour is used at least once and each cycle is either monochromatic or contains at least two red and two blue edges (NAC comes from *No Almost Cycle*, which are cycles in which all edges but one have the same color). It was proven by [5, Theorem 3.1] that a connected graph with at least one edge is the graph of a linkage in the plane if and only if it has a NAC-colouring. The proof is done in two very distinct parts; the first part proves via valuation theory that any linkage induces a NAC-colouring, while the second gives an algorithmic method to construct a linkage from any given NAC-colouring. While the construction given for each NAC-colouring does give a linkage, the linkage will often not share any of the symmetries of the graph it was formed from (see for example Figure 1). A natural question now arises; can we adapt the result so as to preserve any chosen symmetry of our graph?

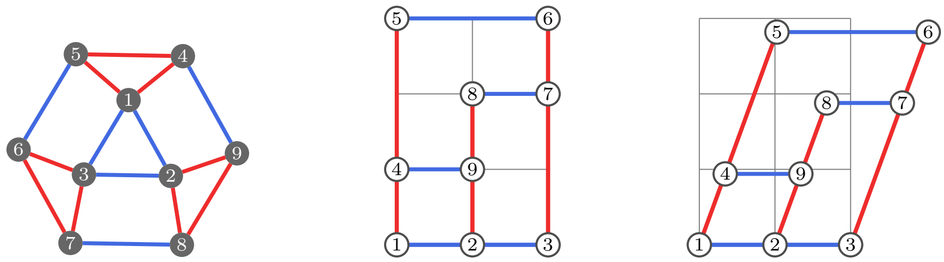

Figure 1 A graph (left) with a linkage constructed from the ilustrated NAC-coloring (middle). The motion of the linkage (right) is parametrised by the angle of the currently vertical lines to the fixed bottom horizontal line.

2 Linkages with rotational symmetry

We define a graph to have *n-fold rotational symmetry* if the group $\mathcal{C}_n := \langle \omega : \omega^n = 1 \rangle$ acts freely on the graph, i.e., each vertex has an orbit of exactly n elements. We define a placement p of G to be *n-fold rotational symmetric* if the placement of a rotated vertex is the rotation of the placement of the vertex, i. e. $p(\omega^k v) = \tau(\omega^k)p(v)$ for each vertex v and rotation $\omega^k \in \mathcal{C}_n$, where $\tau(\omega^k)$ is the rotation matrix for angle $2\pi k/n$.

If this holds then we define (G, p) to be a *n-fold rotational symmetric framework* with *n-fold rotational symmetric placement p*. We further define (G, p) to be a *n-fold rotational symmetric linkage* if there is a continuous edge-length preserving motion of (G, p) that maintains the rotational symmetry but is not a rotation of the framework. Remembering this, we can define the correct type of NAC-colouring to take into account the graph's rotational symmetry.

▶ **Definition 1.** *Let G be an n-fold rotational symmetric graph with NAC-colouring δ. We define δ to be an n-*fold rotational symmetric NAC-colouring *if the colouring respects the symmetry of the graph, and no two distinct blue, resp. red,* partially invariant *connected components are connected by an edge; a set of vertices U is partially invariant if there exists $\gamma \in \mathcal{C}_n \setminus \{1\}$ such that $\gamma U = U$.*

The ideas of n-fold rotational symmetric linkages and n-fold rotational symmetric NAC-colourings tie together nicely similarly to how linkages and NAC-colourings do.

▶ **Theorem 2.** *An n-fold rotational symmetric connected graph with at least one edge is the graph of an n-fold rotational symmetric linkage in the plane if and only if it has an n-fold rotational symmetric NAC-colouring δ.*

The full proof of Theorem 2 comes in two parts and can be found in [1]. We detail below the construction part that builds an n-fold rotational symmetric linkage from an n-fold rotational symmetric graph G with n-fold rotational symmetric NAC-colouring δ:

(1) We first need to label our red and blue connected components in a way that respects the symmetry; we will not, however, need to bother doing this for the partially invariant components as we will see in Step 3. We label the not partially invariant red components as $R_1^0, \dots, R_1^{n-1}, \dots, R_m^0, \dots, R_m^{n-1}$, where $R_j^i = \omega^i R_j^0$ for $0 \leq i < n$ and $1 \leq j \leq m$; similarly, we label the not partially invariant blue components as $B_1^0, \dots, B_1^{n-1}, \dots, B_k^0, \dots, B_k^{n-1}$.

(2) Next, we need to choose base vectors for each of the red and blue components that will determine the shape of the framework. The choice is actually (almost) arbitrary, which fortunately will allow us to pick "nice" vectors. Let $a_1, \dots, a_m$ and $b_1, \dots, b_k$ be our choice of points in the plane with the assumption that $a_j \neq \tau(\omega^i) a_{j'}$ and $b_j \neq \tau(\omega^i) b_{j'}$ for $j \neq j'$ and $1 \leq i < n$. This assumption is necessary to avoid overlapping vertices.

(3) Using our choices of a_i's and b_j's from Step 2, we now create a "coordinate system" in which vertices are placed depending on the red and blue component they belong to. To do this, we define the functions $\overline{a}, \overline{b} \colon V(G) \to \mathbb{R}^2$ by

$$\overline{a}(v) = \begin{cases} \tau(\omega^i) a_j & \text{if } v \in R_j^i \\ (0,0) & \text{otherwise,} \end{cases} \qquad \text{and} \qquad \overline{b}(v) = \begin{cases} \tau(\omega^i) b_j & \text{if } v \in B_j^i \\ (0,0) & \text{otherwise.} \end{cases}$$

We note that a vertex is mapped to the origin by $\overline{a}$ (respectively, $\overline{b}$) if and only if it lies in a red (respectively blue) partially invariant component.

(4) Finally, by using our "coordinate system" determined by $\overline{a}, \overline{b}$ we define for each $t \in [0, 2\pi]$ an n-fold rotational symmetric placement p_t of G, where for each $v \in V(G)$ we have

$$p_t(v) := \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix} \overline{a}(v) + \overline{b}(v)\,.$$

This yields indeed an n-fold rotational symmetric linkage with the corresponding motion given by $t \mapsto (G, p_t)$. Further, we can change the linkage by choosing different a_i's and b_j's. This often allows us to construct "better" linkages by choosing new base vectors; whether "better" entails maintaining some other unspecified symmetry (for example, reflectional), or just a linkage that is more aesthetically pleasing.

▶ **Example 3.** By using our construction we can obtain the n-fold rotational symmetric linkages given in Figures 2 and 3.

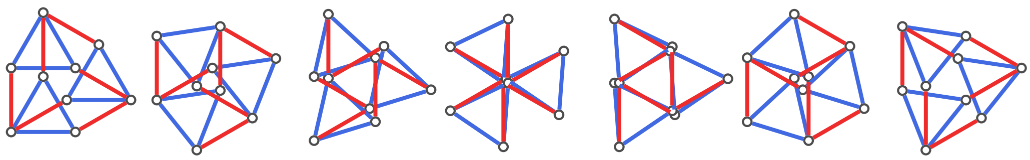

Figure 2 A 3-fold rotational symmetric linkage constructed from a given 3-fold rotational symmetric NAC-colouring.

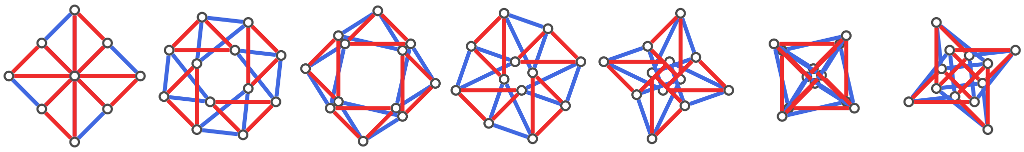

Figure 3 A 4-fold rotational symmetric linkage constructed from a given 4-fold rotational symmetric NAC-colouring.

3 Software for Animations

Animations can be created by an implementation of the above described algorithm using canonical choices for the base vectors. An updated version [4] of the package [3] can be used to study $\mathcal{C}_n$-symmetric frameworks. We encourage the reader to experiment[1] with the choice of a_i's and b_j's from Theorem 2. The implementation can also be used to find the symmetric NAC-colouring. However, in both graphs of Figure 4 we chose a simple NAC-colouring where two triangle subgraphs intersecting in a single vertex are coloured differently. Basic animations can be created with the software packages. The provided animation was constructed using graphical post-processing for the coordinate output.

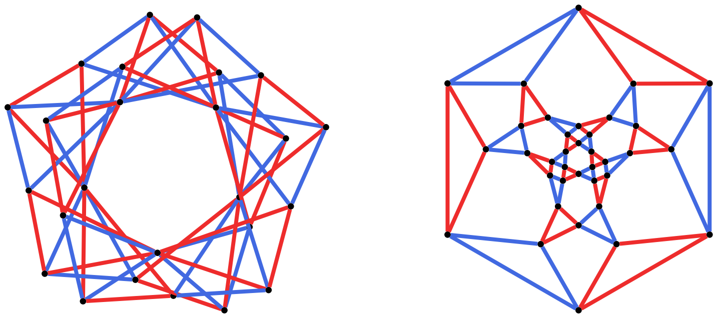

Figure 4 Two graphs with their symmetric NAC-colourings used for the animations. To illustrate the difficulty of finding the symmetric motion, we only show some arbitrary graph layouts. In fact, the animations will never reach either of these layouts.

References

1. S. Dewar, G. Grasegger, and J. Legerský. Flexible placements of graphs with rotational symmetry. Technical report, arXiv, 2020. `arXiv:2003.09328`.
2. A. C. Dixon. On certain deformable frameworks. *Messenger*, 29(2):1–21, 1899.
3. G. Grasegger and J. Legerský. FlexRiLoG – SageMath package for Flexible and Rigid Labelings of Graphs. Zenodo, March 2020. `doi:10.5281/zenodo.3719345`.
4. G. Grasegger and J. Legerský. FlexRiLoG – SageMath package for Flexible and Rigid Labelings of Graphs, repository, 2020. URL: `https://github.com/Legersky/flexrilog/`.
5. G. Grasegger, J. Legerský, and J. Schicho. Graphs with Flexible Labelings. *Discrete & Computational Geometry*, 62(2):461–480, 2019. `doi:10.1007/s00454-018-0026-9`.
6. H. Pollaczek-Geiringer. Über die Gliederung ebener Fachwerke. *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)*, 7:58–72, 1927. `doi:10.1002/zamm.19270070107`.

[1] See file `examples/Rotationally_symmetric_frameworks_SoCGmedia.ipynb` of [4] or try it online: `https://jan.legersky.cz/SoCGmedia2020` redirecting to a Binder version of the notebook.