Routing Reconfigurations

Rebecca Lin
MIT
Cambridge, USA
ryelin@mit.edu

Wenzhong Yan
UCLA
Los Angeles, USA
wzyan24@g.ucla.edu

Ankur Mehta
UCLA
Los Angeles, USA
metank@g.ucla.edu

Erik D. Demaine
MIT
Cambridge, USA
edmaine@mit.edu

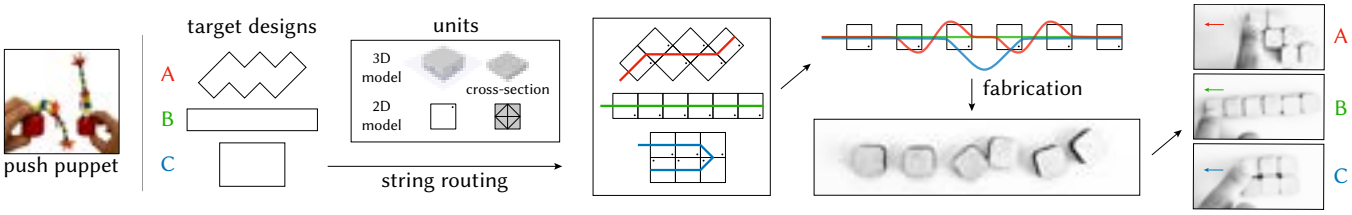


Figure 1: Each string threaded through a series of units forms a unique target shape when pulled taut.

ABSTRACT

We propose an approach to reconfiguration: routing string through a collection of geometric units so that pulling on different strings or groups of strings automatically (re-)arranges the units into distinct target configurations. We provide several strategies for smooth reconfiguration, including computing minimum-turn routings and optimizing the geometry of the units to reduce resistive force.

CCS CONCEPTS

• **Applied computing** → **Computer-aided design**.

ACM Reference Format:
Rebecca Lin, Wenzhong Yan, Ankur Mehta, and Erik D. Demaine. 2024. Routing Reconfigurations. In *ACM Symposium on Computational Fabrication (SCF Adjunct '24), July 07–10, 2024, Aarhus, Denmark.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3665662.3673255

1 INTRODUCTION

The push puppet consists of disjoint, rigid components interconnected by string. It is rigid when the string is taut and becomes flexible when slack is introduced. Our work extends the functionality of push puppets to achieve *multiple* rigid configurations through different string routes (Fig. 1). This work finds application in fields such as space engineering, transformable architecture, and robotics. Specifically, robots can benefit from reconfiguration to adapt to dynamic environments and harness the advantages of both soft and rigid materials [Bern et al. 2022; Yan et al. 2024].

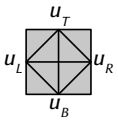
Our Contributions. The main challenge in our approach is designing efficient string routes. Leveraging insights from graph theory, we give algorithms for computing routes through a collection of square building blocks (Sec 3.1). These routes either (1) minimize the total length to reduce material costs and manufacturing time, or (2) minimize the number of turns to decrease friction during deployment.[1] We also discuss insights into the fabrication of our designs (Sec. 3.3) and avenues for future research (Sec. 5).

2 RELATED WORK

Many artists and researchers have studied string-actuated forms: Edmark [2016] transforms hinged geometries between linear and spiral arrangements using a single cable; Kilian et al. [2017] create curved folded surfaces from crease patterns by tightening a network of string; Martin [2021], Lin and Tachi [2024], and Demaine et al. [2024a,b] thread strings through tubes to achieve a target structure when pulling the strings taut. Our work extends these foundations to reach many distinct target configurations. While other works like the snake cube [Abel et al. 2013] and FlexTruss [Sun et al. 2021] leverage threading for reconfiguration, they do not explore integrating multiple threading routes within a single system.

3 METHOD

We employ a single building block: a square unit with holes that form a complete graph on the midpoints of the sides (inset). We route string through these units to assemble these units into target polyominoes when tightened. We first demonstrate this approach given a single target configuration (Sec. 3.1), and then extend the technique to accommodate multiple target configurations (Sec. 3.2).



[1] According to the Capstan equation, the friction between the string and the blocks increases *exponentially* with the sum of the absolute turn angles in the threading route.

SCF Adjunct '24, July 07–10, 2024, Aarhus, Denmark

ACM ISBN 979-8-4007-0695-0/24/07
https://doi.org/10.1145/3665662.3673255

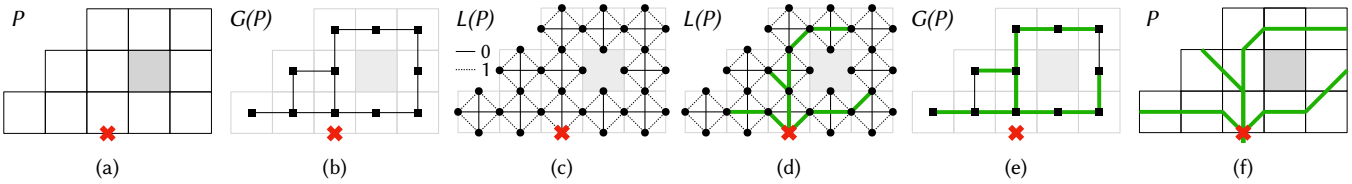


Figure 2: A minimum-turn routing of string (f) for a polyomino (a).

3.1 Computing String Paths

We take as input a pair (P, a_*) that comprises an n-unit polyomino P and an actuation point a_*, where $* \in \{T, B, L, R\}$ indicates the top, bottom, left, or right side of a (Fig. 2a). We employ the grid graph $G(P)$ of P (Fig. 2b) to leverage graph-theoretical insights. Our key observation is that a spanning tree in $G(P)$ provides a practical template for routing strings through P. Below, we discuss approaches to computing spanning trees that either (1) use a single string of minimal length, or (2) use a *collection* of string, each optimized to have as few turns in P as possible.

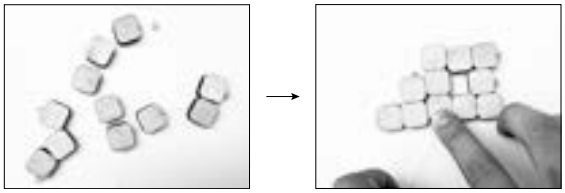
3.1.1 Shortest Path Using a Single String. This task is equivalent to finding a Hamiltonian cycle in $G(P)$, which is efficiently solvable for polyominoes without holes using the algorithm by Umans and Lenhart [1997]. The string traverses the cycle, starting at a_* and terminating with a fixed end at the last unit in the cycle. If $G(P)$ is not Hamiltonian, then we apply the following algorithm.

3.1.2 Minimum-Turn Paths Using Multiple String. Here, we do not limit the number of string. However, the route from the actuation point to each unit must have as few turns as possible.

Let $L(P)$ denote the union of complete graphs on the units of P (Fig. 2c). If unit u is directly left of v, then traversing node uv, or equivalently nodes u_R or v_L, in $L(P)$ indicates a direct routing from unit u to v in P. An edge (wu, uv) in $L(P)$ incurs a cost of 1 if the units w, u, and v form a bend, and 0 if they lie in a straight line.

We apply Dijkstra's algorithm to find the shortest path from the actuation point a_* to one of $\{v_T, v_B, v_L, v_R\}$ in $L(P)$ for each vertex v in $G(P)$ (Fig. 2d). These paths are then mapped into a shortest path tree in $G(P)$ (Fig. 2e), with each root-to-leaf path $p_1, \ldots, p_\ell$ traversed by a string (Fig. 2f).

Assuming a user-specified slack of δ between the units to allow for a loose state and space for reconfiguration, the minimum slack required for string i is then given by $s_i := \delta \cdot (\#p_i - 1)$, where $\#p_i$ is the number of nodes along path p_i. We apply the maximum required slack to each string universally so that all strings are taut simultaneously, as shown in the example below.[2]



[2]Alternatively, instead of precisely measuring the length of each string, a practical heuristic is to first "deploy" the system into the target configuration with each string pulled taut, then cut the strings at a length from the actuation point that approximately allows for the desired amount of slack.

3.2 Achieving Multiple Configurations

Given input pairs $(P_1, a_{*1}), \ldots, (P_k, a_{*k})$, we apply our prior algorithms to each pair individually so that pulling on the jth string, or group of string, from actuation point a_{*j} achieves configuration P_j.

The slack must now account for the distances between units across configurations (Fig. 3). We adjust the required slack for string i in P_j accordingly:

$$s_i' := \max_{m \in [k]} \left(\|p_i\|_{P_m} - \|p_i\|_{P_j} \right) + s_i, \tag{1}$$

where $\|p_i\|_{P_m}$ denotes the shortest routing of the units in path p_i under configuration P_m.

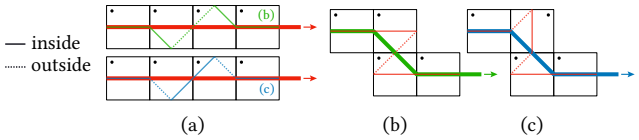
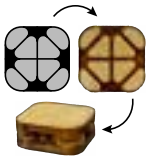


Figure 3: Two ways (b) and (c) to reconfigure a straight line (a) into a staircase. The prior allows for smoother transitions.

3.3 Fabrication



We round all corners of the units to minimize snagging and to reduce friction during rearrangement. These units are produced by engraving and cutting the pattern shown in the inset, and then gluing the pieces together. Our prototypes are made from plywood and nylon string.

4 RESULTS

See our supplementary videos for several results in action. We provide more examples and further discussions on our website: `https://rebeccayelin.github.io/routing-reconfigurations`.

5 FUTURE WORK

Sufficient slack is necessary for reconfiguration, but excessive slack can cause bunching and tangling. To address this issue, we plan to consider common spanning trees of target polyominoes to ensure consistent unit orderings across arrangements. Likewise, the disjoint nature of the units often results in disordered states that impede smooth deployments, so we shall explore hinged dissections for more structured arrangements. We will also further refine our designs for better alignment and expand them to include 3D units and more complex geometries, as our algorithms are adaptable to these scenarios. Finally, we hope to investigate robot-assisted string routing to streamline manufacturing.

REFERENCES

Zachary Abel, Erik D Demaine, Martin L Demaine, Sarah Eisenstat, Jayson Lynch, and Tao B Schardl. 2013. Finding a Hamiltonian Path in a Cube with Specified Turns is Hard. *Information and Media Technologies* 8, 3 (2013), 685–694.

James M Bern, Leonardo Zamora Yañez, Emily Sologuren, and Daniela Rus. 2022. Contact-Rich Soft-Rigid Robots Inspired by Push Puppets. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*. IEEE, 607–613.

Erik D Demaine, Yael Kirkpatrick, and Rebecca Lin. 2024a. Graph Threading. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik.

Erik D Demaine, Yael Kirkpatrick, and Rebecca Lin. 2024b. Graph Threading with Turn Costs. *arXiv preprint arXiv:2405.17953* (2024).

John Edmark. 2016. Roll-Up Spiral. https://www.johnedmark.com/spirals1/2016/4/29/roll-up-spiral

Martin Kilian, Aron Monszpart, and Niloy J Mitra. 2017. String Actuated Curved Folded Surfaces. *ACM Transactions on Graphics (TOG)* 36, 3 (2017), 1–13.

Rebecca Lin and Tomohiro Tachi. 2024. Push Puppet-Inspired Deployable Structure. https://twitter.com/rebeccayelin/status/1749193197031469102

Alison Martin. 2021. Optimization of Threading Paths. https://twitter.com/alisonmartin57/status/1461643652946698240

Lingyun Sun, Jiaji Li, Yu Chen, Yue Yang, Zhi Yu, Danli Luo, Jianzhe Gu, Lining Yao, Ye Tao, and Guanyun Wang. 2021. Flextruss: A Computational Threading Method for Multi-Material, Multi-Form and Multi-Use Prototyping. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

Christopher Umans and William Lenhart. 1997. Hamiltonian Cycles in Solid Grid Graphs. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*. IEEE, 496–505.

Wenzhong Yan, Talmage Jones, Christopher L Jawetz, Ryan H Lee, Jonathan B Hopkins, and Ankur Mehta. 2024. Self-Deployable Contracting-Cord Metamaterials with Tunable Mechanical Properties. (2024).