Stranding Webs

Heather M. Russell[1] and Julianna S. Tymoczko[2]

[1]University of Richmond, Richmond, Virginia, USA; hrussell@richmond.edu
[2]Smith College, Northampton, Massachusetts, USA; jtymoczko@smith.edu

Abstract

Webs visually model structures in the mathematical field of representation theory, with applications across mathematics. We introduce *strandings*, which add color to traditional web constructions, streamlining calculations and uncovering global structures. We show that natural properties of stranding illuminate deep mathematical information about webs, open new lines of research, and benefit from the insights and intuition of design-oriented thinking.

Introduction

Mathematical visualization has the power to spark insight, build intuition, and connect seemingly unrelated concepts [2]. There is a rich history of this approach in the field of knot theory. For instance, quantum knot theorists encode tools from representation theory via graphs called *webs* and apply them to analyze knots. Our work also focuses on visual techniques for extracting data from webs [3, 5].

Traditionally, webs are drawn as graphs whose edges are labeled by numbers—see Figure 1a. Refining the decorations into binary strings that satisfy *additional* technical conditions creates even more cluttered and opaque figures—see Figure 1b—but each binary-string-decorated graph gives a unique term in what's called a *web vector*. Mathematicians then use the web graphs to transform complicated arithmetic operations like addition of the underlying web vectors into graphical operations like twisting and rewiring the graphs [1].

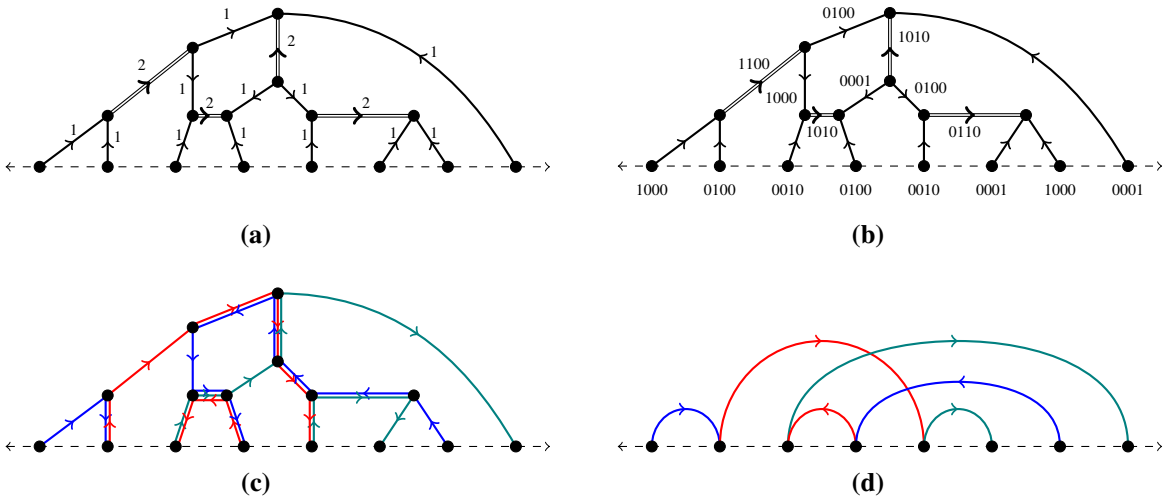

Figure 1: *(a) An $\mathfrak{sl}_4$ web where single edges have label 1 and doubled edges have label 2, (b) a binary labeling, (c) the corresponding stranding, and (d) the underlying colored noncrossing matching.*

The technical conditions that define webs can actually be interpreted as colorings of the entire web graph—like Figure 1c—that we call *strandings*. Strandings eliminate the need for some of the computations in traditional constructions of webs. Moreover, strandings create patterns and designs that lead to entirely new insights, and can be streamlined as in Figure 1d to identify previously-unknown structures in webs.

Describing Webs and Strandings

We begin by defining webs, binary labelings, and strandings, with examples in Figure 1. An $\mathfrak{sl}_n$ *web* is an oriented and weighted graph where, at each vertex above the axis, the difference of incoming and outgoing weights is a multiple of n ("*flow in equals flow out*"). We can picture webs as a network of channels carved into a surface. The edge labels represent the width of each channel and the *flow in equals flow out* condition means the total capacity of all channels is preserved at the junctions where three channels meet.

Traditionally, a web vector is described as a sum of terms, each of which corresponds to a *binary labeling* of an $\mathfrak{sl}_n$ web. A binary labeling is a string of n zeros and ones on each edge with (1) exactly the same number of ones on each edge as its label in the web, and (2) at each vertex above the axis, the sum of the incoming strings minus the outgoing strings has the same value in each entry.

Binary labelings seem to be technical objects defined edge-by-edge. In fact, they represent a beautiful global structure across the web. Pick $n-1$ different colors for the strands in $\mathfrak{sl}_n$ webs and call them $\omega_1, \ldots, \omega_{n-1}$. In our images, we use blue, red, and green for ω_1, ω_2, and ω_3. From a binary labeling, lay an ω_k strand across any edge whose string has different values in its k and $(k+1)^{st}$ entries. Orient the strand with the edge if these entries are 10 and against if the entries are 01. Figures 1c, 3, and 6 show examples.

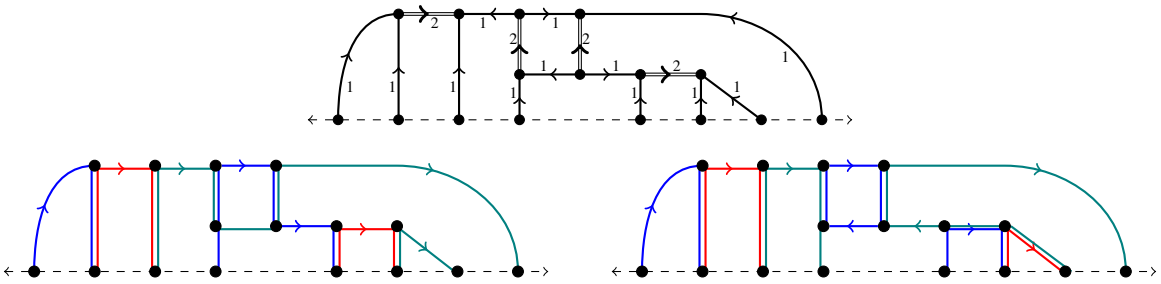

Figure 2: *An $\mathfrak{sl}_4$ web with two different strandings.*

Remarkably, every strand entering a vertex above the axis also leaves that vertex; that two strands of the same color never run along the same edge or pass through the same vertex; that every edge has at least one strand; that each colored strand either starts at the axis and ends at the axis or forms a closed loop within the graph [4]. If webs are channels of various widths in a surface, then the color ω_k has width k, and *stranding the web* is a way of using widths of each colored strand to completely fill all channels, according to the rules we just described. Figure 2 gives two strandings of the same web, created without binary labelings.

Mathematicians want to answer questions like: given a web, is there a better method than guess-and-check to strand it? Can we find all strandings of a web? Are there features that make a stranding better or worse? The rest of this article points to possible partial answers, all of which are motivated by visual aspects of strandings like clockwise/counterclockwise direction, nesting, and symmetry.

How To Find A Stranding: The Base Stranding

The bigger and more complicated a web gets, the harder it is to find even one binary labeling. We have an explicit algorithm that produces a stranding from the input of a web. The stranding that this algorithm outputs is called the *base stranding*. It can be turned into a binary labeling if desired.

Our algorithm has two steps. First, we label the faces of a web from the outside in by the following process. Label the unbounded face 0. Now imagine crossing an edge from an already-labeled face to a still-unlabeled face. If that edge points to your left, add the edge label; if it points to your right, add n minus the edge label. Finally, record this quantity in the new face modulo n. (In other words, if this quantity exceeds $n-1$ then subtract n, as if telling time with a clock that has n hours.) Figure 3 shows this process for an $\mathfrak{sl}_4$

web, where double edges mean the label is 2.

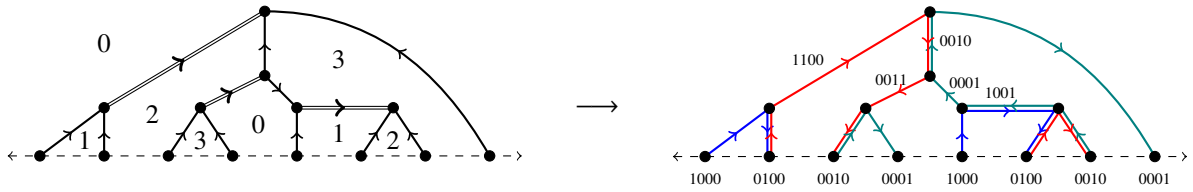

Figure 3: *The two steps to create a base stranding for an $\mathfrak{sl}_4$ web, with binary labeling for comparison.*

In the second step, run a strand clockwise along all edges bounding each face, using color ω_k if the face was labeled k in the first step. (Faces with label 0 contribute nothing to the stranding.) We can prove the resulting stranding is valid in the sense that follows all the rules from the previous section, and indeed encodes a binary labeling. Interestingly, if all edges at the boundary of an $\mathfrak{sl}_4$ web are pointed up and labeled 1 then the binary labels along the boundary of the base stranding repeat $1000, 0100, 0010, 0001$, and similarly for any n. Mathematically, this shows that all webs with these boundary edges have a common term in their web vectors. Many other patterns—mathematical and visual—can likely also be detected from strandings.

Depth and Nesting: Completing Strandings and Measuring Complexity in Webs

Now suppose we choose a binary labeling for the boundary edges of a web, like the repeated sequence $1000, 0100, 0010, 0001$ above, or like Figure 4. When can we find a binary labeling of the entire web extending this choice? Stranding helps answer this, too.

Figure 4: *A binary labeling and strand configuration, with faces labeled by the number of strands above.*

For instance, the binary labels on the boundary in Figure 4 tell us exactly where strands start and end. Strands of the same color do not cross, so we can only join these fragments in one way. Since the right side of Figure 4 is connected, it cannot possibly be used to strand any web with multiple disconnected pieces.

Depth is another visual property that helps test if a web admits a stranding. The *depth* of a web face is the least number of web edges one must cross to get to the unbounded face, if crossing the dashed axis or through vertices is forbidden. Figure 5 labels faces of a web with their depths. (These are **not** the face labels from the base stranding!) If a web is stranded, we must cross all strands lying above a face before we get to the unbounded face. Figure 4 shows a strand configuration whose faces are labeled with the number of strands lying above each region.

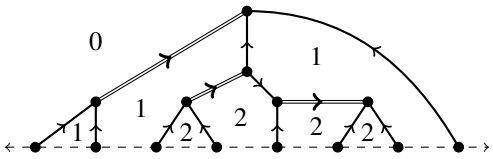

Figure 5: *Face depths in an $\mathfrak{sl}_4$ web.*

In other words, each face in a stranded web is at least as deep as predicted by its "strand nestedness." This means the web in Figure 5 cannot be stranded with the pattern from Figure 4 since the depth between the second and third vertices in the web is smaller than between the same pair of vertices in the strand configuration. In short, strandings with more nesting can only be laid onto webs that are more complex.

Finding More Strandings: Strand Flipping

Stranding also allows us to create new binary labelings from existing ones. To *flip a strand* that is colored ω_k, both (1) reverse its direction and (2) for each edge on the strand, toggle whether or not it has ω_{k-1} (and similarly ω_{k+1}). Flipping the leftmost ω_2 strand in Figure 3 gives Figure 6: instead of the first edge of the strand being blue, the last four are; and similarly for the green edges. Flipping a strand actually swaps the k^{th} and $(k+1)^{st}$ entries in the binary labels of all edges along the strand.

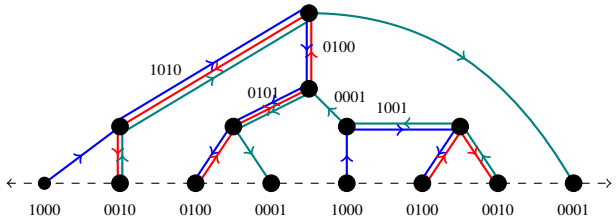

Figure 6: *The stranding and binary labeling obtained from flipping the first red strand in Figure 3.*

Flipping a strand may shift, reconnect, create, or destroy other strands from the original stranding. Starting at the base stranding, we can flip strands repeatedly to reach other strandings of the same web. We conjecture that all strandings are in fact reachable from the base stranding via some sequence of strand flips.

Summary and Conclusions

This article argues that the technical and seemingly-local binary labelings of webs are better visualized as colored *strands* that fill the entire web. This change of perspective gives new and natural ways to measure and think about webs, including nestedness and clockwise/counterclockwise. Stranding also opens new questions, e.g. which blends of colors hold mathematical significance. We believe that using stranding, people trained in design can provide insights that would fuel tremendous progress on research in webs.

Acknowledgements

The first author is supported by an AMS-Simons Research Enhancement Grant for PUI Faculty. The second author is supported by NSF DMS-2054513. Both authors thank the Budapest Semesters in Mathematics DMiR Program for support in summer 2023 that was invaluable to this work, and are deeply grateful for many inspiring conversations with Felicia Flores, Emily Hafken, Veronica Lang, and Orit Tashman.

References

[1] G. Kuperberg. "Spiders for Rank 2 Lie Algebras." *Communications in Mathematical Physics*, vol. 180, no. 1, pp. 109–151, 1996. https://doi.org/10.1007/BF02101184

[2] H. M. Russell and R. Sazdanovic. "Mathematics and Art: Unifying Perspectives." *Handbook of the Mathematics of the Arts and Sciences*, pp. 497–525, 2021. https://doi.org/10.1007/978-3-319-70658-0_125-1

[3] H. M. Russell and J. Tymoczko. "The Transition Matrix Between the Specht and $\mathfrak{sl}_3$ Web Bases is Unitriangular with Respect to Shadow Containment." *International Math Research Notices*, no. 5, pp. 3371–3416, 2022. https://doi.org/10.1093/imrn/rnaa290

[4] H. M. Russell and J. Tymoczko. "Stranding $\mathfrak{sl}_n$ Webs." In Preparation, 2024.

[5] J. Tymoczko, "A Simple Bijection Between Standard $3 \times n$ Tableaux and Irreducible Webs for $\mathfrak{sl}_3$." *Journal of Algebraic Combinatorics*, vol. 35, no. 4, pp. 611–632, 2012. https://doi.org/10.1007/s10801-011-0317-1