Visualizing the Computational Complexity of Knots and Links

Shannon Cheng and Saarah Nazar

Abstract

We are investigating the computational complexity of two decision problems - the Independent Set Problem and the Trivial Sublink Problem. We provide a reduction from the former to the latter. This gives a new, simpler proof that the Trivial Sublink Problem is NP-hard. Our project consists of two main goals: a careful write-up of this new streamlined proof, and a visualization tool that implements the reduction, thereby giving users intuition on why problems in link theory are computationally hard.

Background and Motivation

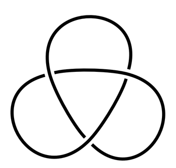


Figure 1: Trefoil knot

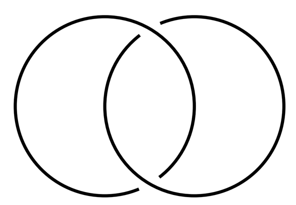


Figure 2: Hopf link

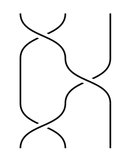
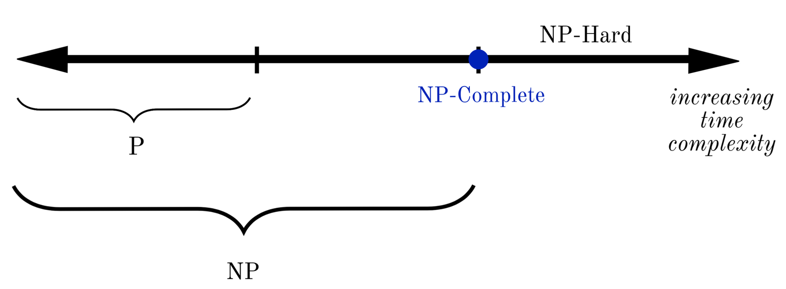
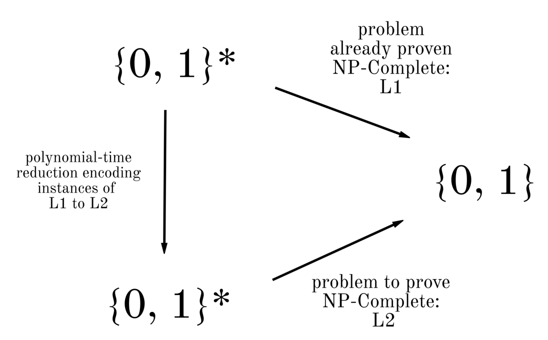


Figure 3: A 3-strand braid

A **knot** is an embedding of circle into $\mathbb{R}^3$. Knots can be represented with 2d **knot diagrams** as in Figure 1. A **link** is a generalization of a knot that allows for several individual knots (called *components*) that can be linked around one another as in Figure 2. **Braids** can be used to represent both knots and links in a simplified, standardized way in terms of which strands cross over which other strands. An example can be seen in Figure 3. To convert a braid diagram to a link diagram, we "close" the braid by connecting the top end of each strand to the bottom end of the strand directly below it.

ɡorized by their time complexity,
t size.





The motivation for this project is to provide a simple-as-possible example that exhibits the intrinsic computational complexity of knots and links. To this end, we first identified a specific decision problem about links called the Trivial Sublink Problem, and we then showed it is NP-hard by reducing from the Independent Set Problem (a well-known NP-complete problem). To aid in intuition, we also built a visualizer.

Trivial Sublink Problem

A **trivial sublink** is a subset of k components that are completely unlinked and with each component unknotted- that is, they can be separated in 3D space without any entanglement or crossings between them. These loops are topologically equivalent to disjoint, unknotted circles.

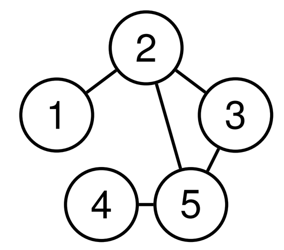
The **Trivial Sublink Problem** asks: Given a link diagram D and an integer k, does D contain a trivial sublink with exactly k components?

Independent Set Problem

A **(simple, undirected) graph** $G = (V, E)$ consists a finite set of *vertices* $V = \{1, \ldots, n\}$ and a set of *edges* $E \subset V \times V$ with the properties that $(j, i) \in E$ for all $(i, j) \in E$, and $(i, i) \notin E$ for any $i \in V$. A graph can be represented by a $n \times n$ symmetric square matrix with 0 and and 1 entries—called the **adjacency matrix**—where $A[i][j] = 1$ means that there exists an edge between nodes i and j. An **independent set** of G is a subset $I \subseteq V$ such that no two vertices in I are connected by an edge.

The **Independent Set Problem** asks: Given a graph G (via its adjacency matrix) and an integer k, does there exist an independent set of size k?

For example, consider the following graph and its adjacency matrix:



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Suppose $k = 3$. Is there an independent set of size $k = 3$? Let us systematically search:

1. {1, 2, 3}: not independent - 1 and 2 are adjacent, 2 and 3 are adjacent
2. {1, 2, 4}: not independent - 1 and 2 are adjacent
3. {1, 2, 5}: not independent - 1 and 2 are adjacent
4. {1, 3, 4}: INDEPENDENT! - no edges between 1, 3, and 4

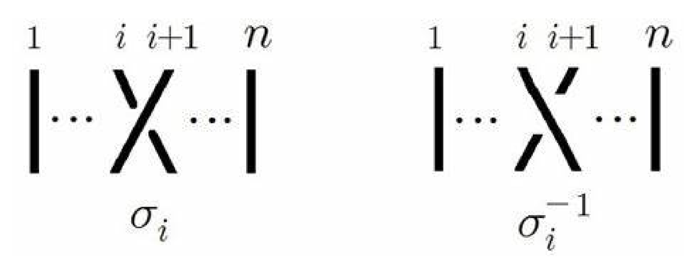
This example suggests that finding independent sets may somtimes require checking many combinations. In fact, the Independent Set Problem is NP-hard, implying (if the Strong Exponential Time Hypothesis is true) that there does not exist any subexponential time algorithm to solve it.

Reduction

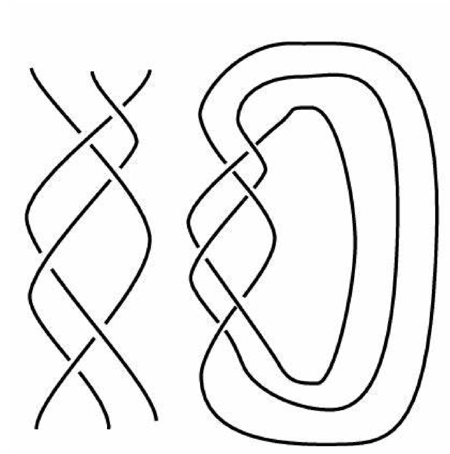
We perform a reduction from the Independent Set Problem to the Trivial Sublink Problem using braid words. We first encode a graph's adjacency matrix A into a braid word w_A, then close it to form a link diagram L_A. Here is the explicit construction:

$$w_A \overset{\text{def}}{=} \prod_{i=0}^{n} \left(\prod_{j=i}^{n-2} \sigma_j \cdot \prod_{j=n-2}^{i} \sigma_j^{-cos(\pi * A[i][j])} \right)$$

where

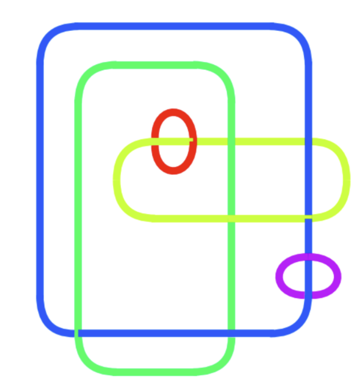


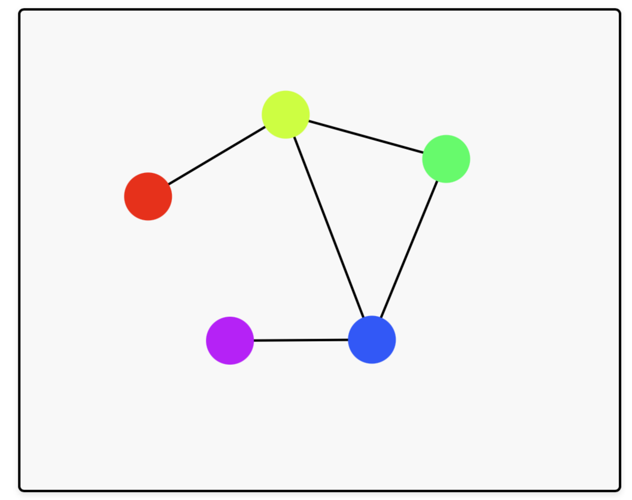
σ_i σ_i^{-1}



b CL(b)

Here is an example of what the reduction looks like using our visualization tool: **https://trivial-sublink-git-main-shannonc8s-projects.vercel.app/**





Key Property of the Reduction

Claim: For all $k \in \mathbb{Z}_{>0}$, L_A has a k-component trivial sublink if and only if A has an independent set of size k.

Proof sketch: Consider a k-component sublink $L' \subseteq L_A$ such that for each pair of subcomponents $K_1, K_2 \subseteq L'$, $\text{lk}(K_1, K_2) = 0$, i.e. their linking number is 0. We want to show L' is trivial.

Each K_i corresponds to a strand i in the braid diagram b_A. Consider arbitrary strands n and m such that $n < m$ and delete the other $k - 2$ strands of b_A. This allows us to relabel our strands, namely $n = 1$ and $m = 2$. Notice that:

$$\begin{cases} \sigma_1^1 \sigma_1^1 & \text{if } A[n][m] = 1, \\ \sigma_1^1 \sigma_1^{-1} & \text{if } A[n][m] = 0. \end{cases}$$

When two components are not linked, their two corresponding strands can be simplified to become trivial using Reidemeister-2 moves.

Acknowledgments

We would like to thank our graduate mentor Anna Natalie Chlopecki and Professor Samperton for all their kindness, mentorship, and guidance with this project.

References

[1] Arnaud de Mesmay, Yo'av Rieck, Eric Sedgwick and Martin Tancer. The unbearable hardness of unknotting. Advances in Mathematics, 381:107648, 2021.
[2] Dale Koenig and Anastasia Tsvietkova. NP-hard problems naturally arising in knot theory. Transactions of the American Mathematical Society, 8(15):420-441, 2021.
[3] Marc Lackenby. Some conditionally hard problems on links and 3-manifolds. Discrete & Computational Geometry, 58:580-595, 2017.