Coarse Grained Networks

John Tobin, Dylan Laurianti
Mentor: Professor Eikmeier

What is a Network?

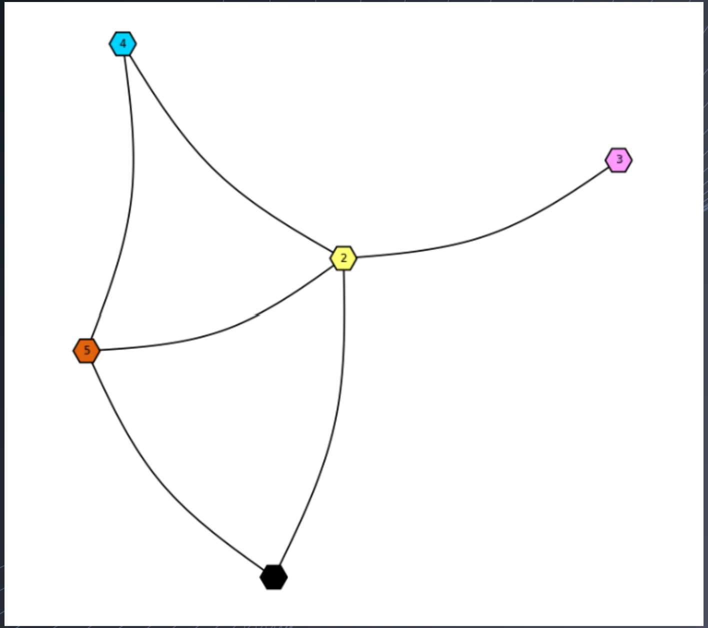
A network, sometimes called a graph, is a collection of nodes and edges.

Nodes are sometimes called vertices.

An edge connects two nodes to each other, or one node to itself.

Many different real-world situations can be represented with networks. For example, nodes could represent people, and an edge between two nodes could signify that those two people know each other.

We have many different ways to generate networks that give them predictable properties.

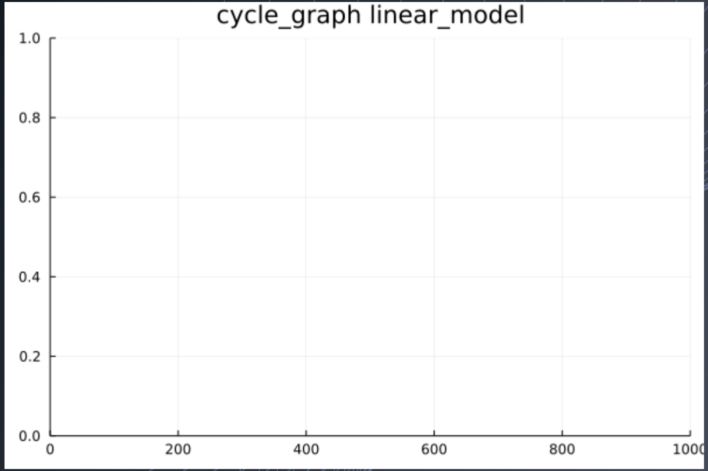


Networks Help Represent Dynamical Systems

Modeling dynamics on networks uses nodes, edges, and node variables to model the change in a system over time.

We use differential equations to compute the change in each node's variable as influenced by the nodes it shares edges with.

There are many types of dynamical models. We're using linear dynamical models, the SI and SIS models, the kuramoto model, and more.



Smaller Networks are Faster Networks

The time it takes to run dynamical processes on networks increases with the number of nodes.

Our goal is to reduce the number of nodes while maintaining as much dynamical behavior as possible.

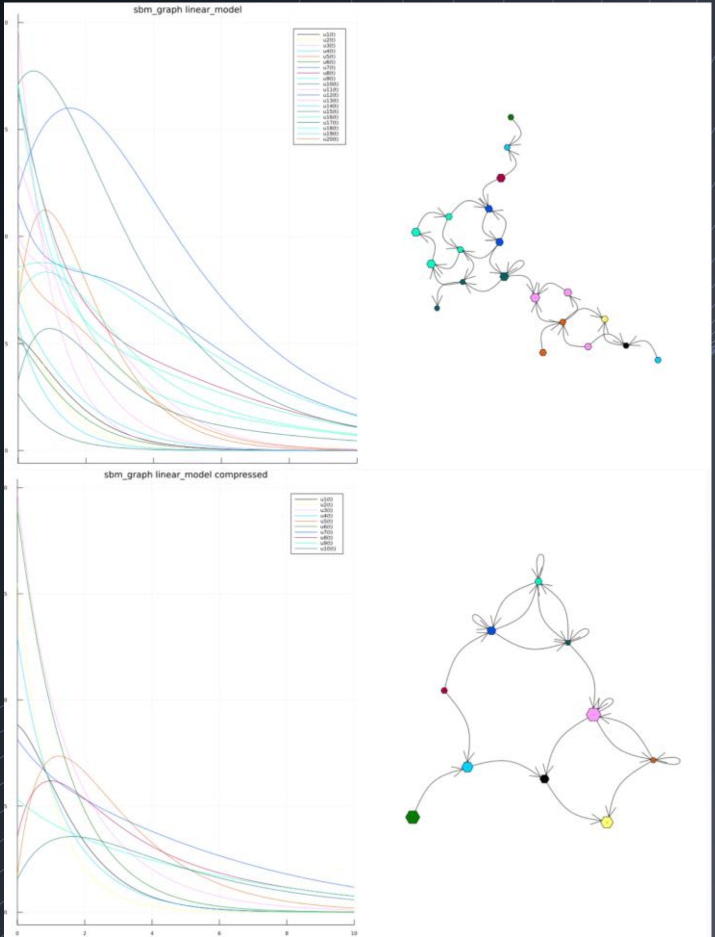
The technique we're using to do this is called Coarse Graining.

N = 20

Slow

N = 10

Faster

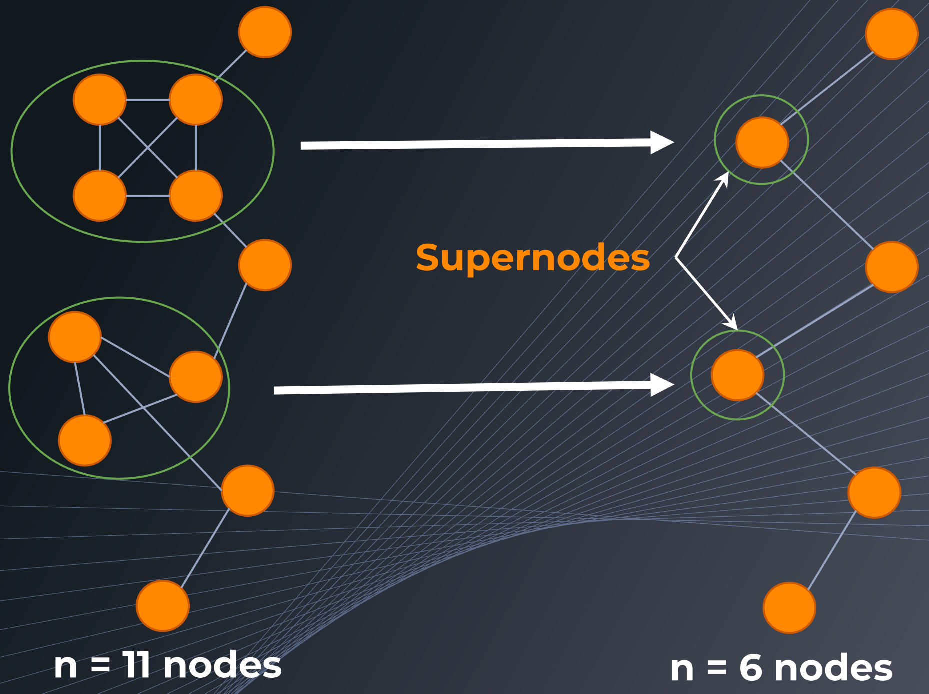


How Does Coarse Graining Work?

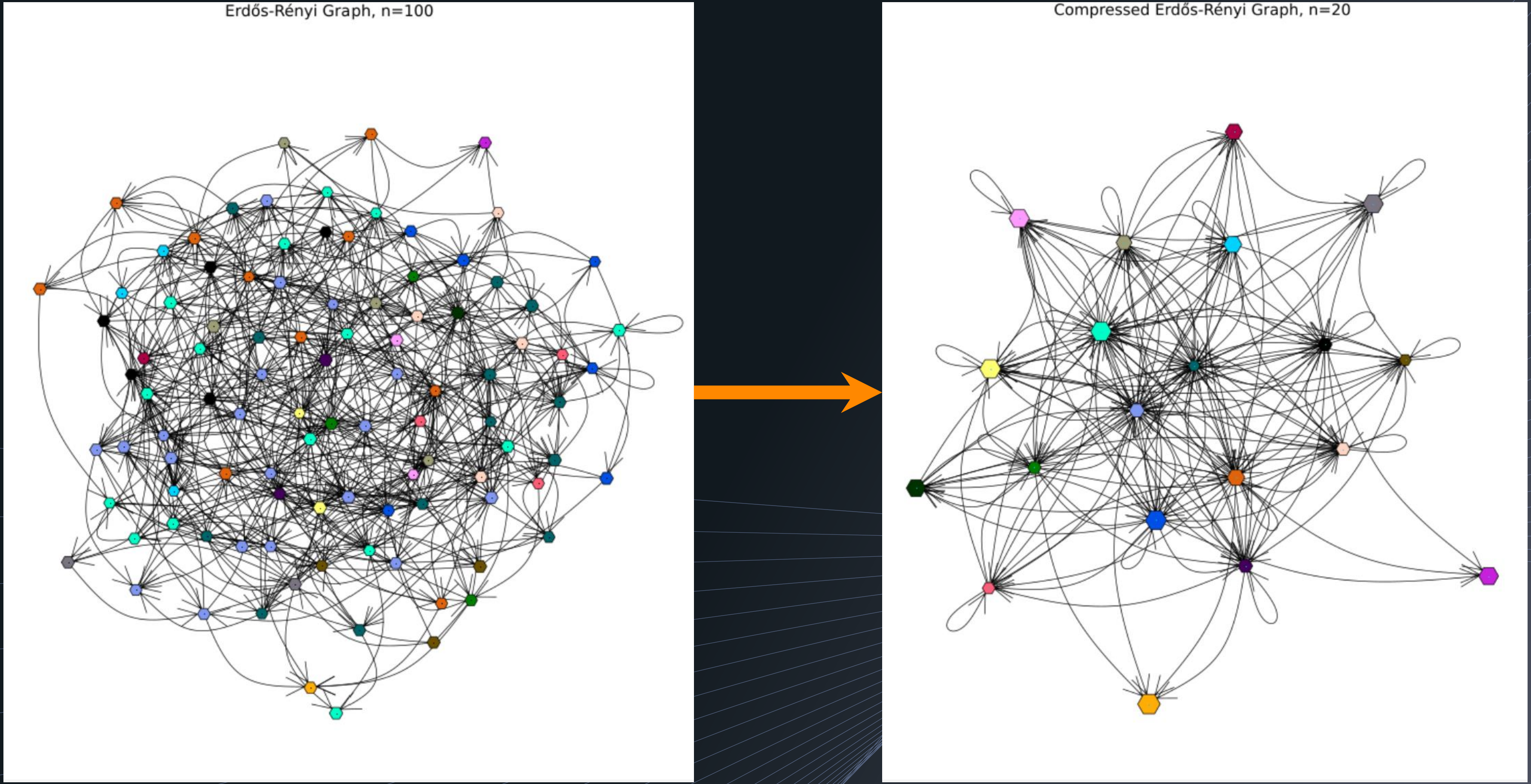
The idea behind Coarse Graining is to combine nodes in a network into supernodes

This is done by making a partition of the original network

Example partition from n=11 to n=6



Example Coarse Grained Network

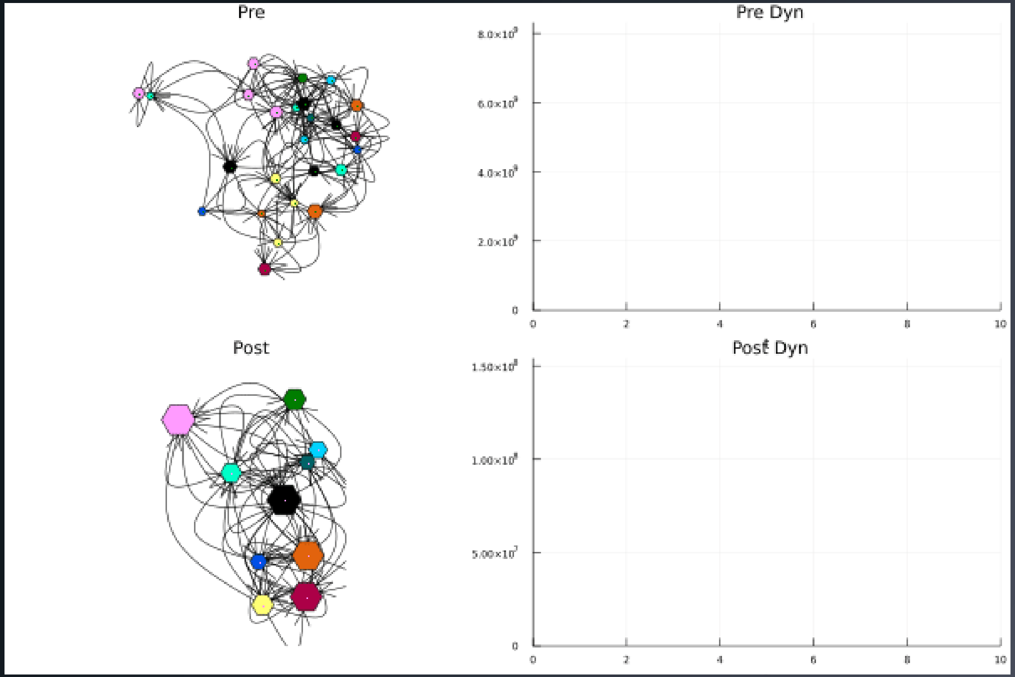


Coarse Graining Causes Loss

These networks would have different dynamical behavior

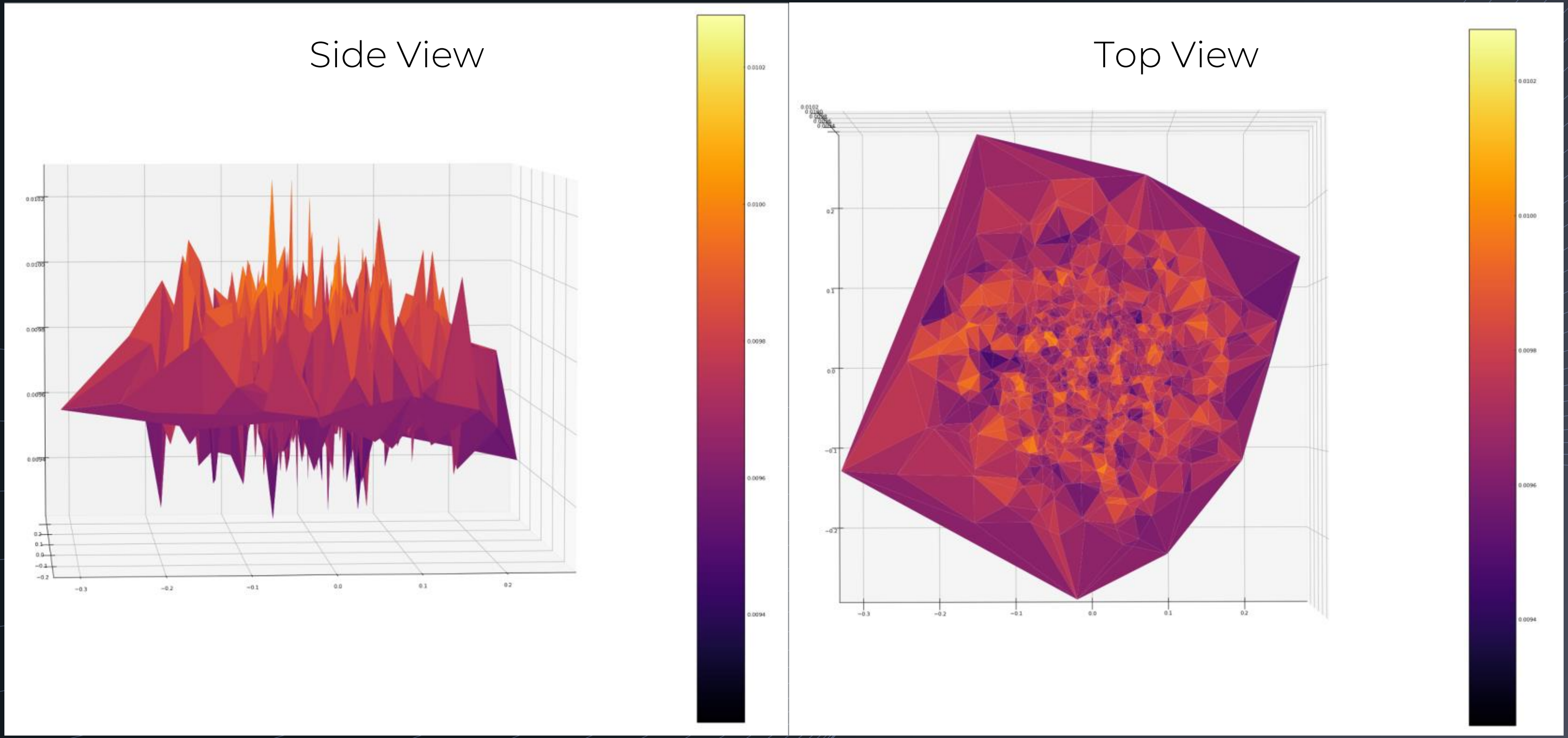
Loss is a measure of the difference between the dynamical behavior of two networks.

Our goal is to find ways to make partitions that minimize loss



Plotting Loss Landscape

Loss landscape of a stochastic block model network with 200 nodes reduced to 100 nodes across 1000 partitions using the SIS model



Side View

Top View

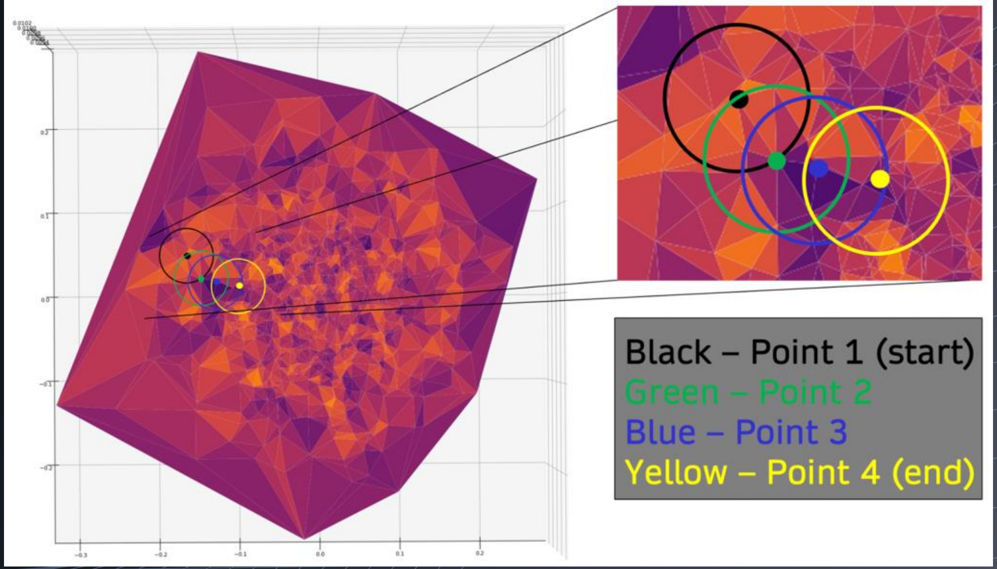
Finding Good Partitions

Optimization on Partition Spaces

1. Generate an original network, a sample of partitions of that network, and run a dynamical model on the network to generate a loss landscape.

1. Run a gradient descent optimization algorithm on the loss landscape to find local minimums of the sample space.

1. Run a combinatorial optimization algorithm using the local minimum of the sample space as input to find approximate local minimums of the partition space.

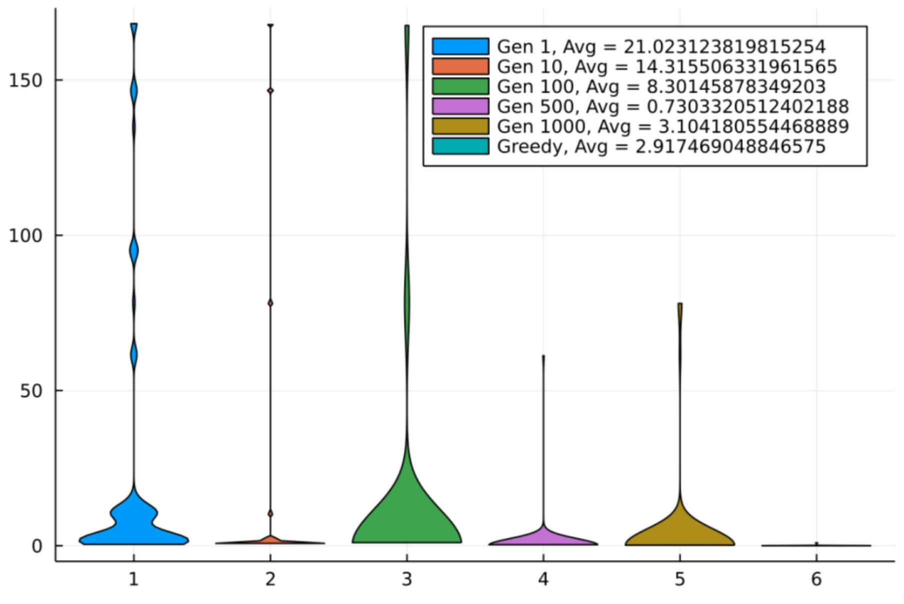


Combinatorial Optimization

Optimization on Partition Spaces

Because the partition space is discrete with high dimensionality, optimization is approached through combinatorial optimization heuristics.

We have currently coded two options for the combinatorial optimization algorithm, an iterative greedy approach which applies gradient descent to the discrete partition space, and a genetic algorithm which reproduces, crossbreeds, and mutates partitions based on their loss-fitness.

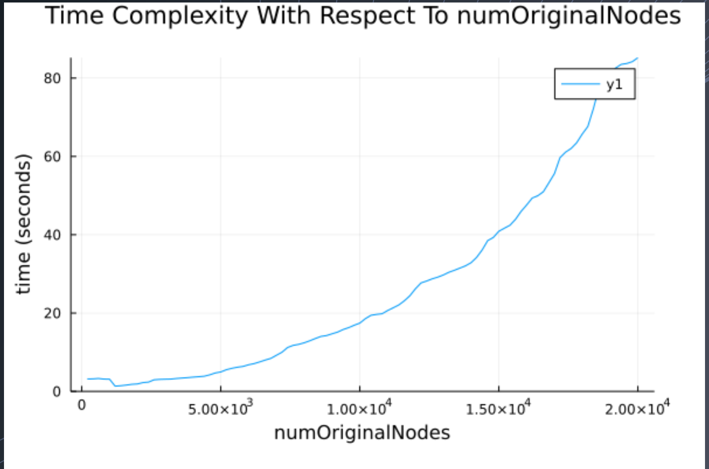


Real World Networks Present a Computability Problem

The complexity and size of some real-world networks rules out the use of dynamical models.

However, dynamical models are some of the strongest tools we have to analyze these networks.

If we can find ways to reduce the number of nodes in such real world networks while preserving dynamical behavior, using dynamical systems on networks could become much more practical because they would no longer be time-prohibitive.



Progress Overview

So far, we have:

- Translated the original Python into Julia, and applied optimizations
- Enabled parallelization
- Added functionality for partition optimization, plotting, and data storage.
- Run experiments exploring the way that partitions change as they are optimized
- Developed methods of visualizing and computationally analyzing partitions

Next we're going to:

- Continue running large-scale experiments
- Develop analytical ways of recognizing patterns in good partitions
- Use them to make better partition networks with certain properties.
- Expand our methods of partition creation and optimization to include more complicated algorithms.

Thank You!

Acknowledgments

Thank you to Professor Eikmeier and her collaborators Alice Schwarze, Nicholas Landry, Phil Chodrow, Mari Kawakatsu, and Benji Zusman