Google
Summer of Code

Proposal: GSoC 2023

Agamdeep Singh

Organisation: NumFOCUS - NetworkX

NetworkX
Network Analysis in Python

NUMFOCUS
OPEN CODE = BETTER SCIENCE

March 2023

1 Project Details

Project Goal: Incorporating a Python library for ISMAGS isomorphism calculations

1.1 Abstract

NetworkX is a python library for complex network modelling, graph representations and studying graph theory. It provides users with a modular interface to create graph and network representations and optimised algorithms to study their properties, dynamics and interactions. It also comes with the functionality to visualise and identify patterns and substructures within graphs, which is what ISMAGS aims to do.

ISMAGS is a **graph isomorphism checking** algorithm that solves the subgraph matching problem. NetworkX already has an implementation of multiple graph isomorphism algorithms here including ISMAGS. Sandia Labs has converted the original ISMAGS algorithm, written in Java by its authors to Python here. This GSoC project aims to understand the differences between the two implementation approaches, assimilate the best subroutines, and combine them to create an **optimised implementation for ISMAGS.**

1.2 Technical Details

Given a graph G and subgraph SG, a subgraph matching algorithm aims to find all instances of SG in G. This is done by:

1. For each node sg_k in SG, find candidate nodes C_k in G that can be mapped to sg_k. Candidate nodes are selected by pruning out nodes in G that don't have the same type of incoming and outgoing edges as $sg1$.

2. Map a node g_k from C_k to sg_k.

3. For each node sg_s adjacent to sg_k, if it is connected by edge type B, prune its candidate set C_s so that each node is connected to sg_k by edge type B.

4. Do this for each neighbour of each neighbour....

5. Iteratively map a new candidate node to sg_k and repeat from step 3.

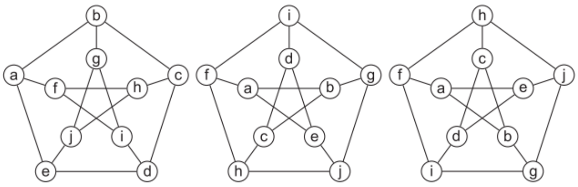
6. By iteratively mapping and backtracking, all subgraph instances are found.

As you can see, this algorithm is exponential in nature, as for each subgraph node, we expand a new tree for each candidate node.

ISMA(Index-based subgraph matching algorithm) was designed to solve the subgraph matching problem faster by taking into account symmetries within the graph. This allows it to identify and skip mapping nodes that are symmetric and replaceable by other nodes it has already expanded over.

While ISMA performs well for small subgraphs with reflectional and rotational symmetry, it fails to tackle larger subgraphs with more elaborate symmetric substructures. This was overcome by ISMAGS(Index-based subgraph matching algorithm with general symmetries), which can take into account larger, more complex symmetric substructures by using partitioning of nodes, coupling, recursive refinement, orbit pruning, stabilisers, coset representatives and symmetry-breaking constraints.

Figure 1: The Peterson graph is a graph with complex symmetries. The 3 permutations are of the same graph



The goal of this project is to compare the current implementation of ISMAGS in NetworkX with the one by Sandia Labs and find the optimal combination of both. This can be done as a modification of the original NetworkX implementation, by wholly adopting the Sandia Labs implementation(if it proves to be better in all aspects) or a new package that combines the best from both implementations.

The possible improvements in the current NetworkX implementation, curated after going through the code:

1. The algorithm **only works with directed graphs.** No such requirement is there in the paper.

and conversing with the contributors here:

1. Improving subroutine that forms node compatibility dictionary between a subgraph and graph.
2. Use certain data structures that will optimise speed. eg. code where one of the contributors pointed out using a sorted set will be faster than a set.
3. Efficient implementation seems to fail here as commented by contributor link

This list is of course not exhaustive. As my understanding of the paper and both the implementations(NetworkX, Sandia Labs) increases, along with a discussion with the mentors, I am confident new issues and scopes for improvement will come to light.

1.3 Scope of contribution

Since the project entails improving on the implementation of existing code, the contribution/improvements can be divided into two classes:

- **Algorithmic**: Improvements in the pseudo-code of the subroutines. These changes will aim for drastic improvements in the complexity of the subroutines by changing their flow. It also includes swapping NetworkX code with Sandia Labs code and vice-versa, choosing the one with better running time complexity.
- **Data structural**: Improvements gained by changing various data structures used throughout the code. This can be done once the pseudo-code is fixed and is an implementation change.

I will aim first to identify and implement all algorithmic improvements and then implement all data structural improvements.

Testing

Code development has to be supplemented with rigorous testing. Our testing must cover two aspects: mathematical completeness and correctness, and compatibility with the rest of NetworkX.

I plan to supplement the Sandia Lab's doctests with additional cases to ensure mathematical completeness. Further, mathematical proof of completeness and correctness on the final implementation will also help. Since we are adding a few features(eg directed graphs), doctests have to go beyond the current NetworkX implementation.

2 Project Timeline

April 4 - May 4 **Pre-GSoC** Gain familiarity with the **mathematical concepts** and algorithm in the ISMAGS paper. Understand requisite graph theoretic concepts. Devise my own pseudo-code for ISMAGS. This will allow me to have a solid foundation to understand what both the implementation aims to do on an algorithmic level.

May 4 - May 14 **Community Bonding - 1.1** Familiarise myself with the structure of **NetworkX implementation** and discuss it with mentors. Calculate the time complexity of subroutines and functions. Gain familiarity with the different NetworkX functionalities as suggested by mentors and interact with the community to understand the structure of the library. First blog report!

May 15 - May 28 **Community Bonding - 1.2** Familiarise myself with **Sandia Lab implementation.** Calculate the time complexity of subroutines and functions. Deliberate on differences compared to Networkx implementation with mentors. Identify algorithmic improvements. Decide whether to form a new package or modify the existing ISMAGS implementation. Blog report 2, 3.

May 29 - Jul 10 **Development Phase 1** Start **implementing algorithmic improvements.** By the end of this period, we should have a new, improved implementation with quite a few of algorithmic changes implemented. Compare running time with the original implementation and Sandia Lab implementation by running tests. By now, the *improved* code should be performing at least as well as the original implementation(barring data structural improvements). Submit code for review. Blog report 4

July 11 - 14 **Review Phase 1** Create **first pull request.** Discuss algorithmic bottlenecks and possible data structural optimizations in my code with mentors. Receive feedback from the rest of the NetworkX community. Blog report 5

July 14 - Aug 1 **Development Phase 2.1** **Remove bottlenecks** discussed in midterm evaluation. Implement remaining algorithmic improvements, if any. Discuss and finalise data structural changes to improve the running time of code with mentors. Blog report 6, 7.

Aug 1 - Aug 21 **Development Phase 2.2** Implement **data structural changes** and measure improvement. Supplement code with extensive documentation and examples as is customary in NetworkX code. Write unit-tests. Blog report 8, 9

Aug 22 - Aug 28 **Review Phase 2** Discuss final changes with mentors. Receive style guide and review feedback from the community, implement changes where necessary. Package code with all **documentation and tests.** Submit for final evaluation. Discuss further improvements to be made after GSoC. Blog report 10(final)!

3 Personal Details

3.1 Contact Info and profile

- Name: Agamdeep Singh
- email: agamdeep20@iiserb.ac.in, agammessi10@gmail.com(alternate)
- Mobile: +91 9417574801
- GitHub: github.com/jnash10
- LinkedIn: linkedin.com/in/agamdeep-iiser
- Location: Bhopal/Chandigarh, India
- Timezone: UTC/GMT +5:30 hours, IST(Indian Standard Time)

3.2 Education

- University: Indian Institute of Science Education and Research Bhopal
- Degree: Bachelor of Science in **Data Science and Engineering**
- Current Year: 3rd
- Current grade: 8.71/10

3.3 Skills

- Fields: Algorithms, Machine Learning, Network Science, Graph Theory, Number Theory
- Languages: Python(6+ Years), MySQL
- Mathematics: Graph theory, Group theory, Linear algebra, Number theory, multivariable calculus, probability and statistics
- Libraries: NetworkX, OSMnx(OpenStreetMap + NetworkX), matplotlib, numpy, pandas, scikit-learn, TensorFlow
- Version control: Git
- Build systems: CMake, Makefile
- Database: MySQL
- Others: Object Oriented Programming, Dynamic Programming, ROS(Robot Operating System), Tableau, Deep Learning, Computer Vision, NLP

3.4 Experience

3.4.1 Open Source

- SageMath (merged) - sagemath/sage : added check for invalid range in contour_plot and derivatives #35113

Solved an edge case in error handling while plotting using the contour_plot base class. This was my first real contribution to open source, something I had been fascinated by for quite long. I got to interact and go back and forth with my reviewer and learned about coding conventions and style guides via this contribution. I was quite happy when it finally got merged.

3.4.2 Research and development

- **Intern - Indian Institute of Science**
  Created SQL database for data backend. Mapped raw GPS coordinates onto Bangalore roads via efficient MySQL queries and OSMnx(**NetworkX** graph models for OpenStreetMap). Wrote scripts for querying and visualising graph networks.

- **Undergraduate Researcher - Multi-Robot Autonomy Lab, IISER Bhopal**
  - Developed a reactive algorithm for priority-based collision avoidance among UAVs(Unmanned Aerial Vehicles). Wrote code for and ran simulations. Research paper under writing.
  - Created a solution to reduce overcrowding in dining halls via wifi fingerprinting of mobile devices.

3.4.3 Other Relevant experience

I am a runner's up in two national-level hackathons. You can find my project links here:

- Pravega 2021 : Efficient-ration-delivery-in-post-disaster-scenario
- Hackduino 2021 : Unfone (look through rubble)

I also love using **NetworkX** for my academic projects. The following was a submission for my Data Science in practise course, where I modelled how social media recommender systems can affect our mental health. Can Social Media make you depressed? : A multi agent simulation I used a probabilistic model for forming new connections and relied solely on **NetworkX** for my modelling and visualisation(along with some matplotlib of course!).

3.5 About me

I am a mathematician who loves solving problems via modelling and programming. This lead me to pursue Data Science and Engineering as my major, where I get to work at the intersection of mathematics, network science and machine learning. I wholeheartedly enjoy collaborating with people, especially online, where I get to interact with specialised experts. The desire to present my best work forward gives me unbound motivation to work through relentless nights. I also love documenting my work, as it gives me a chance to showcase how things work and why they are designed the way they are. These interests took me to different academic labs in the past and now to GSoC and, more specifically, NetworkX, which serves as a culmination of all my interests. Aside from programming, I play for my university Football(soccer) team, head the Computing and Networking Council, IISERB as a student representative and dabble in robotics occasionally.

3.6 Commitment to GSoC and NetworkX

I have no prior commitments during the GSoC'23 timeline. My academic semester and final exams will be over before the timeline starts, and I will wholeheartedly devote myself to GSoC and NetworkX full-time, spending 40-50 Hours per week. I will be conversing daily with my mentor regarding updates, doubts, approaches, and code through the assigned mode of communication. I will also compile each week's progress into a blog post, which I will post weekly.

4 Motivation

4.1 Why Google Summer of Code

Since I started programming in the 9th grade, open source has fascinated me. Over the past few years, I was lucky enough to gain experience programming in different fields - algorithms, computational mathematics, machine learning and data engineering. Throughout this process, I was continually exposed to different programming concepts and tools that challenged me and improved my programming practises when I mastered them. I was also fortunate to apply these skills by contributing to various research laboratories in India and fell in love with collaborating through code. This experience gave me the confidence to tackle large open-source repositories and submit my first PRs. Now I can finally say I'm ready to take on GSoC as a full-time summer project, equipped with the necessary tools and skills to succeed and make meaningful contributions, taking open science and open source forward.

4.2 Why NetworkX

My first introduction to NetworkX was via a senior at my university, whom I was helping with some visualisation for his master's thesis. He was modelling road congestion between different bus stops in Delhi and planning routes. The conversation that ensued after I enquired about the library he was using opened for me the world of network science and modelling, and at the backbone of it all was - NetworkX. I then followed NetworkX tutorials and started using it for my academic and research projects. Currently, NetworkX serves as the first choice for graph and network modelling by teachers teaching DSA, researchers modelling the brain or studying how societies interact and mathematicians looking for visualisation for their results. Being able to contribute to such an organisation will be akin to a dream come true and boost my confidence in impacting the world through my code.

4.3 Why this project

This project offered me the perfect opportunity to tickle my mathematician bone while providing a meaningful contribution to NetworkX. The idea of scouring through different parts of the NetworkX and the Sandia Labs implementation, comparing and searching for optimal subroutines keeping the original paper as a reference provides the perfect research challenge. Coding the combined implementation also excites me as I can draw from the best of both world and write my own code, like an inspired poet. Learning from the different approaches taken by the contributors, for each good approach, I will have the other that isn't so good, leading me to better appreciate what makes the good approach, well, *good*. I will be exposed to good design guidelines, and coding practises while studying the implementations and will get to use them when I combine them or write my own code.

4.4 Why choose me

The current skillset I possess equips me perfectly to take on this project. From my mathematical background in graph and group theory to my development and open-source experience contributing to SageMath. Add to that research and collaboration experience in different research labs where I worked with new codebases, developed new algorithms and presented my work weekly, documenting it for other researchers. I am highly motivated to work on this project and you can expect the utmost discipline in terms of deadlines, deliverables, blog reports and my work ethic. I plan to continue contributing to NetworkX beyond GSoC. Hopefully, one day igniting the open-source torch in some other student's mind, the way NetworkX has done for me.