# Application for Internship with NetworkX

Project: Implement new parallel algorithms in nx-parallel

# Past experience with this community:

Please describe your experience before this Outreachy application period with this free software community. You can describe your prior experiences as both a user and a contributor.

Almost two years back, I did a course on Social networks by Prof Sudarshan Iyengar where I first got introduced to the NetworkX library and what we can do with it. To get a better understanding of the course content I implemented some social computing models using Networkx(including k-shell decomposition, PageRank, Schelling's model, etc.) and notes for the theoretical

parts(<u>https://notability.com/n/iF~vsBG\_3Ut4vZoNURzNc</u>) and I also curated a list of learning resources, and relevant papers, etc.

(https://github.com/Schefflera-Arboricola/Stuff/tree/main/Projects/Social%20Computing%20models). And, I didn't have any prior experience as a contributor in the NetworkX community. I started contributing to it in the Outreachy contributing stage.

\_\_\_\_\_\_

# Past experience with other communities:

Please describe your experience before this Outreachy application period with any other free software communities. You can describe your prior experiences as both a user and a contributor.

I have been a user of many free open-source softwares for a long time and I recently started contributing actively to them, when I participated in GirlScript Summer of Code 2023. Before that, I was just developing my own projects on GitHub. In GSSoC, I contributed to a web scraping project[1]. Later I also used that project for my own project(i.e. creating a network dataset of developers and repos on GitHub[2]). I also worked on a Computer vision project[3], but I only contributed to the documentation, not to the CV part. I was also part of the CodessCafe OpenSource Mentorship cohort 2023 where I got to interact with a lot of people working in open-source software communities and former interns of open-source programs like Outreachy, GSoC, MLH, etc., which helped me gain some very useful insights. In both programs, I got to meet a lot of like-minded and helpful peers who I am really grateful for.

### GSSoC'23 merged PRs:

[1]

https://github.com/Clueless-Community/scrape-up/pulls?q=is%3Apr+author%3ASchefflera-Arboricola

[3] https://github.com/akshitagupta15june/Face-X/pull/1538

#### [2] GitHub network dataset:

https://www.kaggle.com/datasets/aditijuneja/github-bipartite-graph-datasetdevelopersrepos

\_\_\_\_\_

# Relevant Projects:

Please describe any relevant projects (either personal, work, or school projects) that helped you gain skills you will use in this project. Talk about what knowledge you gained from working on them. Include links where possible.

Worked on the following during the Outreachy contributing stage ->

- 1. Parallel implementation of all\_pairs\_bellman\_ford\_path(embarrassingly parallel) algorithm: <a href="https://github.com/networkx/nx-parallel/pull/14">https://github.com/networkx/nx-parallel/pull/14</a>. By working on this, I got a deeper understanding of how generators work and how we can parallelize the functions that return generators. I am hoping to close this one before the internship period starts.
- 2. Implemented skeleton benchmarking infrastructure: <a href="https://github.com/networkx/nx-parallel/pull/24">https://github.com/networkx/nx-parallel/pull/24</a>. While working on this I got familiar with how big GitHub projects are set up and organized with tests and workflows and automated docs using Sphinx and how the projects are configured and built using CircleCl's config file. Here I used ASV for benchmarking. I still have a lot to learn but I got an overall idea about setting up repositories by working on this.

Other contributions made: All these contributions helped me understand the NetworkX repository better.

- 1. [OPEN] <a href="https://github.com/networkx/networkx/pull/7027">https://github.com/networkx/networkx/pull/7027</a> : added examples in core.py
- 2. [MERGED] <a href="https://github.com/networkx/networkx/pull/7013">https://github.com/networkx/networkx/pull/7013</a> : updated docs of SA tsp and TA tsp
- 3. [MERGED] <a href="https://github.com/networkx/nx-parallel/pull/13">https://github.com/networkx/nx-parallel/pull/13</a> : bug fix : changed edge probability from 0.5 to p
- 4. [OPEN] <a href="https://github.com/networkx/networkx/pull/6995">https://github.com/networkx/networkx/pull/6995</a> : Addition in docs of the traveling\_salesman\_problem() function about triangle inequality case.
- 5. [MERGED] 4PRs -

https://github.com/networkx/outreachy/pulls?q=is%3Apr+author%3ASchefflera-Arboricola: 5 initial Outreachy tasks(the heatmap PR helped me understand how the CPU cores work and also helped me getting started with exploring the nx-parallel repository)

#### Blog on TSP approximation

algorithms(https://schefflera-arboricola.github.io/Schefflera-Arboricola/TSP-approximation-algorithms): based on what I learned while exploring the TSP algorithms in the networkx repo.

\_\_\_\_\_\_

# Outreachy internship project timeline:

Please work with your mentor to provide a timeline of the work you plan to accomplish on the project and what tasks you will finish at each step. Make sure take into account any time commitments you have during the Outreachy internship round. If you are still working on your contributions and need more time, you can leave this blank and edit your application later.

Week 1 (Dec 4 - Dec 10) and Week 2 (Dec 11 - Dec 17):
Learn about networkx dispatching and networkx backend and implement
this(nx.betweenness\_centrality(G, backend='nx-parallel')) type of function calls for
parallel graphs, as briefly touched upon in this
comment(https://github.com/networkx/nx-parallel/pull/24#discussion\_r1371014988\_)

Week 3 (Dec 18 - Dec 24):

Improve benchmarking infrastructure to incorporate all kinds of algorithms and graph types. And add benchmark classes for existing algorithms in the nx-parallel project.

Week 4 (Dec 25 - Dec 31):
Add more utility functions: like briefly touched in this comment(<a href="https://github.com/networkx/nx-parallel/pull/14#pullrequestreview-1678180">https://github.com/networkx/nx-parallel/pull/14#pullrequestreview-1678180</a> 807 )

Week 5 (Jan 1 - Jan 7):

Other improvements: like setting up docs(or importing docs(to avoid redundancy) from the main networkx repo), improving workflows or CI, etc.

Week 6 (Jan 8 - Jan 14) and Week 7 (Jan 15 - Jan 21):

- Parallel implementations(+benchmarking+docs+tests) of algorithms in cores.py(https://github.com/networkx/networkx/blob/main/networkx/algorithms/core.p y ) and euler.py(https://github.com/networkx/networkx/blob/main/networkx/algorithms/euler.p y )

- Parallelizing the \_core\_subgraph function will help in improving the performance(especially if the graph is large) of k\_core, k\_shell, and k\_corona algorithms.(https://github.com/networkx/networkx/blob/main/networkx/algorithms/core.py#L141)

- and a part of the k\_crust algorithm can also be parallelized in a similar way(<a href="https://github.com/networkx/networkx/blob/main/networkx/algorithms/core.py#L3">https://github.com/networkx/networkx/blob/main/networkx/algorithms/core.py#L3</a> 14 ).
- is\_eulerian and eulerize functions can be easily parallelized, and some parts of has\_eulerian\_path, eulerian\_circuit, eulerian\_path functions can be easily parallelized too.

Week 8 (Jan 22 - Jan 28) and Week 9 (Jan 29 - Feb 4):

- Parallel implementations(+benchmarking+docs+tests) for the BFS algorithms(This algorithm might change as discussed with the mentor.)
- References: <a href="https://en.wikipedia.org/wiki/Parallel\_breadth-first\_search">https://en.wikipedia.org/wiki/Parallel\_breadth-first\_search</a>, <a href="https://indoorg/doi/10.1145/3434393">https://indoorg/doi/10.1145/3434393</a>, <a href="https://joblib.readthedocs.io/en/latest/parallel.html#shared-memory-semantics">https://joblib.readthedocs.io/en/latest/parallel.html#shared-memory-semantics</a> (Still researching on this.)

Week 10 (Feb 5 - Feb 11) and Week 11 (Feb 12 - Feb 18):

- Parallel implementations(+benchmarking+docs+tests) for "all\_pairs\_" shortest paths algorithms and algorithms with "# TODO This can be trivially parallelized" comment.
- Most of the "all\_pairs\_" type of algorithms on this page(<a href="https://networkx.org/documentation/latest/reference/algorithms/shortest\_paths.">httml</a>) can be easily parallelized like the all\_pairs\_bellman\_ford\_path(<a href="https://github.com/networkx/nx-parallel/pull/14">https://github.com/networkx/nx-parallel/pull/14</a>). The "# TODO" comment algorithms:
  - boruvka mst edges:

https://github.com/networkx/networkx/blob/f9170cc404bd4bd1c482e976f8f7677b1b6 0a1cd/networkx/algorithms/tree/mst.pv#L122

- global reaching centrality:

https://github.com/networkx/networkx/blob/f9170cc404bd4bd1c482e976f8f7677b1b6 0a1cd/networkx/algorithms/centrality/reaching.py#L112

- node\_redundancy :

https://github.com/networkx/networkx/blob/f9170cc404bd4bd1c482e976f8f7677b1b6 0a1cd/networkx/algorithms/bipartite/redundancy.py#L92

- There are a lot of generators(<a href="https://github.com/networkx/networkx/tree/main/networkx/generators">https://github.com/networkx/networkx/tree/main/networkx/generators</a>) also that can be embarrassingly parallelized.

Week 12 (Feb 19 - Feb 25):

Any other improvements in the project structure or algorithms(like adding README files, etc.) and setting up the documentation website for the nx-parallel plugin, etc.

Week 13 (Feb 26 - Mar 1): Blog post summarising everything I did and learned during the internship.

#### Notes:

- 1. If I will have extra time left at the end:
- Parallel implementation for TSP approx. algorithms(greedy, Christofides algo): for greedy\_tsp, we can run multiple iterations of greedy\_tsp in parallel and then return the best one. This probably won't reduce the time but it could give a better approximation(it will have a little different benchmarking script).
- Parallelising the k\_core algorithm by having a shared memory and then maybe we can run it for different clusters(or disconnected subgraphs) in a graph and then combine the results at the end. I will have to research more on this.(reference: <a href="https://dl.acm.org/doi/pdf/10.1145/3087556.3087580">https://dl.acm.org/doi/pdf/10.1145/3087556.3087580</a>)
- Implement other centrality measure algorithms
- 2. If I would be short on time due to some other thing taking more time than expected then I would prefer to create good-first-issues for some of the Weeks 10-11's trivial algorithms.
- \* discussions/meets with the mentor/team for all the weeks and Outreachy chat meets every 2 weeks
- \* This is a tentative timeline.

Exam dates(during weeks 3 and 4):
19th dec, 22nd dec, 24th dec, 27th dec, 29th dec 2023
(9:30 AM to 12:30 PM)

#### Application Google docs link:

https://docs.google.com/document/d/1xLr4\_kqxWU3dB2GB5AAsSvr3VDtKMssQs5i YoxJhffg/edit?usp=sharing (for any updates in the timeline)

Thank you :)			

# (Optional) Community-specific Questions:

Some communities or projects may want you to answer additional questions. Please check with your mentor and community coordinator to see if you need to provide any additional information after you save your final application.

Email: \*\*\*\*

GitHub: https://github.com/Schefflera-Arboricola

Blog website: https://schefflera-arboricola.github.io/Schefflera-Arboricola/

LinkedIn: https://www.linkedin.com/in/aditi-juneja-940838204/

\_\_\_\_\_

FIN.