Incremental Shortest Path Algorithm

Rutgers University-New Brunswick

Data Structures and Algorithms - 16:332:573

Prof. Dov Kruger

Group - 9

Team Members:

NAME	NET ID	EMAIL ID	GITHUB ID
SUHAS KOLLUR	sk2870	sk2870@scarletmail.rutgers.edu	https://github.com/suhaskollur
BEOMSEOK PARK	bp593	bp593@scarletmail.rutgers.edu	https://github.com/BaamPark
SHUYUAN FAN	sf850	sf850@scarletmail.rutgers.edu	https://github.com/zetsunozomi

GitHub Repository: [BaamPark/IncrementalShortestPathAlgorithm (github.com)](https://github.com/BaamPark/IncrementalShortestPathAlgorithm)

Project Description

We propose to work on implementing and optimizing an incremental shortest-path algorithm inspired by Raghav Daga's concept. The goal is to develop a solution capable of efficiently updating the shortest paths in a graph when the data changes incrementally, such as variations in airfares or train fares in a transportation network.

We will create a fictitious dataset representing costs between various points in our transportation network. Using this dataset, we will implement the Floyd-Warshall algorithm to compute the initial shortest paths between all pairs of vertices. Following this, we will devise and implement an incremental algorithm that updates the shortest paths efficiently when the graph data changes.

Milestones

1. **Literature Review and Algorithm Selection (Week 1):**
   - Conduct an in-depth literature review to identify existing algorithms for incremental shortest paths.
   - Analyze and compare the strengths and weaknesses of various algorithms.
   - Select the most suitable algorithm as the basis for our implementation.

2. **Setup and Dataset Creation (Week 1):**
   - Set up the development environment, including necessary libraries and tools.
   - Create a graph representation of the transportation network.
   - Generate a fictitious dataset of costs between various points in the transportation network.

3. **Floyd-Warshall Implementation (Week 2):**
   - Implement the Floyd-Warshall algorithm to compute the initial shortest paths between all pairs of vertices in the graph.
   - Validate the correctness of the Floyd-Warshall implementation using test cases.

4. **Incremental Update Algorithm Design and Implementation (Week 3-4):**
   - Design the incremental update algorithm based on the selected approach.
   - Define data structures and algorithms necessary for efficiently updating shortest paths with incremental changes.
   - Conduct unit testing to ensure each component of the algorithm functions as expected.

- Integrate the incremental update algorithm with the existing Floyd-Warshall implementation.

5. **Algorithm Optimization (Week 5):**
   - Identify potential bottlenecks and areas for optimization in the incremental update algorithm.
   - Implement optimizations to improve the algorithm's efficiency and scalability.
   - Benchmark the optimized algorithm against the initial implementation to measure performance improvements.

6. **Testing and Evaluation (Week 5):**
   - Develop a comprehensive testing strategy to evaluate the correctness and performance of the implemented algorithms.
   - Perform rigorous testing using various test cases, including synthetic datasets and real-world scenarios.
   - Analyze the results to assess the algorithms' effectiveness in handling incremental changes and compare their performance against each other.

7. **Documentation and Presentation (Week 5):**
   - Document the implementation details, including algorithm descriptions, data structures used, and optimization techniques employed.
   - Prepare a final paper following IEEE format, summarizing the project's objectives, methodology, findings, and conclusions.
   - Create a presentation highlighting key aspects of the project, including the problem statement, approach, results, and future work.

Background Research

We have found several papers with algorithms about incremental shortest-path problems. We will put them into our GitHub repository, read them, determine their relevance, and implement several algorithms from those papers. In the future, we will add more papers to our GitHub repository.

Single-source shortest path:
https://arxiv.org/pdf/2001.10751.pdf
https://theory.stanford.edu/~virgi/cs267/papers/rz-esa04.pdf
https://research.cs.wisc.edu/wpis/papers/jalg96.pdf
All pairs the shortest path: https://publikationen.bibliothek.kit.edu/1000052742/3801174

Responsibilities

Suhas

- Develop and implement the original Floyd-Warshall algorithm on the dataset.
- Create the dataset of fictitious transportation costs for initial algorithm testing.

Beomseok

- Design and code the incremental update algorithm for changing costs.
- Test and compare the performance of both algorithms with varying data sizes.

Shuyuan

- Find papers about incremental shortest path algorithms, and collect algorithms.
- Implement and test other existing algorithms for comprehensive comparison.

The project repository will be hosted on GitHub as a private repository. We will invite the course instructor (hydrodog) and the grader (Deevesh) to the repository to monitor our progress.

This project offers an exciting opportunity to explore both theoretical concepts in graph algorithms and practical implementation challenges in real-world scenarios. We are committed to working collaboratively to achieve the project goals within the specified timeline.

We appreciate your consideration of our proposal and look forward to your feedback.