Abstract

The focus of this project was finding the most reliable path across a complex network. The main requirement for the result was that it had to have a time-complexity similar to that of Dijkstra’s algorithm. The approach taken was to implement two solutions: an implementation of Dr. Yuan’s algorithm and heuristics based solution. Data from both solutions were compared, and confirmed that Dr. Yuan’s algorithm was more efficient.

Background & Introduction

In recent years Network Reliability has become increasingly important. In many communities data is distributed over a network of computers with varying connections and reliabilities. To maintain the flow of information, a specific path fails, the next most reliable path must be found. This is even more important in rural communities where Internet and/or data connections are received on one device and then relayed across the entire network.

To tackle this problem, our research focuses on an efficient solution with a complexity near that of the Dijkstra’s algorithm. The task at hand was divided such into a research phase and an implementation phase. Chronologically speaking the first portion of the project was completed in Summer I 2008 and the latter portion was completed over Summer II.

Methods & Details

Summer I: Research

The first portion of the project consisted of studying complex networks. Several speculative sound algorithms were tested and yet proved to be inefficient. The most promising of the methods observed was Djikstra’s Algorithm.

Dijkstra’s Algorithm

Initial assessments revealed that this particular method works well with simple weighted graphs. However, given more complex graphs and reliability evaluation methodologies the algorithm fails to perform correctly.

For example, in some our test graphs edge weights were interdependent meaning that a given percentage of reliability, all edges of similar reliability were prone succeed and fail simultaneously. Additionally, the weights used were more statistical in nature and relied on laws probability rather than constant values that could be summed.

Although enhanced versions of the algorithm could be considered, since the original algorithm was proven to be fundamentally ineffective for the given task. As a result we opted for more complex methods.

Summer II

The second segment of the project, which started in Summer II involved actually implementing different solutions and assessing sample outputs of each. Such benchmarks allow us to understand both how well each method works individually in addition to which type of data each one was more suited to.

Heuristic Algorithm

The first step is to initialize link reliability to 1 on all links in the network. The next step is to find a maximum-reliability lightpath (using Dijkstra’s Algorithm). Then, pick one color at a time, and temporarily set the link reliability to 1 on all links of that color, and find a new maximum-reliability path. Each color represents a unique reliability. Therefore, links of the same color have the same reliability. Repeat this procedure for all colors in the network, and select the color that results in the path with the maximum number of colors.

Then set the reliability equal to 1 on the links of that selected color. Finally, repeat the second and third steps until the number of colors on the maximum-reliability path cannot be further reduced.

Yuan’s Algorithm

This algorithm has a complexity of O(nColors * 2^nColors – 1). The approach requires the partial population of a table reflecting the paths between the nodes of interest. The following points should be taken into consideration when implementing this algorithm.

To allow for reasonable flexibility in graph size and data management a "vector<vector<node>>" data structure was used.