Algorithms 2500 – Section 1A
Second Project

Deadlines: 12/7 First report and Second report

Recovery of communication networks after large scale failures

Maximum points 100

General information

Maximum group size 3.

Any programming language can be used for the implementation.

The project reports should be submitted in hardcopy to the Instructor the day of the deadline. Additionally, the reports and the code should be submitted to the TA electronically the day of the deadline.

The project should be discussed with the TA within 1 week from the deadline of the second report.

Project goals

Communication networks are today at the basis of our everyday life. Their pervasive distribution and use, makes them a critical infrastructure, which is used to connect critical services such as police stations, hospitals, banks, transportation services, etc.

Natural disasters or intentional attacks, may significantly destroy part of the network, and prevent it to provide connectivity between critical services in such an emergency situation.

The goal of this project is to design and compare the performance of recovery strategies to restore the network functionality.

Network model and problem formulation

Communication networks can be represented using undirected graphs. In our project we consider one node as source of information and another node as the destination. These represent the critical services that need to communicate.

Nodes have coordinates in a 2D space, while links can be modeled as the straight line that connects the two endpoints. Links in the network have a maximum capacity. The capacity of each link should be randomly selected in the interval \([1,10]\).

When a large scale failures occurs in the network several links and nodes are damaged. We consider two types of failure:

- Random: a link or a node can fail with probability \(P\).
We assume that time is slotted in intervals. Since nodes and links may be in different areas, some more inaccessible after the disaster, nodes and links have different repairing time. In particular, the repairing time is expressed in number of time slots and should be randomly selected in the interval [1,100].

**Recovery algorithms**

You should develop a recovery algorithm to schedule the repairs of node and links to restore the flow between the source and the destination. The algorithm eventually recovers the entire network, but the order in which the recovery is performed has impact on the maximum flow over time between the source and the destination.

The goal of your algorithm is to maximize the maximum flow during the recovery process, hence it needs to carefully select the nodes and links to be repaired. As a result, the output of the algorithm is the order of nodes and links to be repaired.

As a baseline comparison, you should implement a random recovery algorithm, that iteratively randomly selects an unrepaired element to be repaired.

**Implementation and evaluation**

You should download the topology of Kentucky Datalink available at this link:

http://www.topology-zoo.org/files/Kdl.gml

You should develop a parser to read the file and import it in your program. Nodes have latitude and longitude coordinates, that should be used to determine the node positions. Nodes also have ids, which can be used to determine the endpoints of the links.

You should perform two sets of experiments, one with random failures and the other with geographic failures.

The choice of the source and destination nodes should be done such that they are far apart in the graph, e.g. considering the nodes with the longest shortest path.

**Random failures**

You should perform two experiments.

In the first experiment, you should consider 4 scenarios with \( P = 0.2, 0.5, 0.7 \) and 1. For each value of \( P \), execute the recovery algorithms to generate the list of nodes and links to be repaired. Given such list, you should calculate, for each instant \( T \), the current max flow with the repaired made until time \( T \).

Generate a graph for each value of \( P \), comparing the random solution with your algorithm. The graphs should have on the X-axis the value of \( T \) and the Y axis the maximum flow at time \( T \). You should produce 4 graphs, one for each value of \( P \).
In the second experiment, you increase the value of $P$ from 0 to 1, step 0.1. For a given value of $P$, let $T_{\text{max}}(P)$ be the time necessary to repair all failed elements. Execute the recovery algorithms and calculate the maximum flow between the source and the destination at time $T_{\text{max}}(P)/2$.

You should produce a graph in which you compare the random and your recovery algorithm. The X-axis should have the value of $P$, while the Y-axis should have the maximum flow at time $T_{\text{max}}(P)/2$.

Overall, 5 graphs should be produced.

Geographical failures

You should perform similar experiments than in the random failure case.

First experiment. Let $D$ be the physical diameter of the network (i.e. the diameter of the smallest circle centered at the barycenter of the network and including all nodes in the network). You should consider 4 scenarios in each scenario all nodes at a distance $R$ from the barycenter fail. The distance $R$ should be set respectively to 0.2$D$, 0.5$D$, 0.7$D$ and $D$.

Generate four graphs comparing your solution to the random solution as explained in the previous case.

Second experiment. Increase the distance $R$ from 0.1$D$ to $D$, step 0.1D. Provide a graph in which you compare the random solution with your algorithm. For a given value of $R$, calculate $T_{\text{max}}(R)$ and the plot the max flow at time $T_{\text{max}}(R)/2$.

Overall, 5 graphs should be produced.

Reports

Two reports should be prepared. The first report does not require implementation and should include at least the following sections:

- Abstract: Summary of the following sections
- Introduction and motivation: Overview of communication network recovery problem
- Proposed solutions: Pseudo-code, description and complexity analysis of the proposed algorithm
- Plan of experiments: Description of the methodology which is planned to use in the experiments, significance and expected results.

The second report requires implementation and describes the implementation and results. It should include at least the following sections:

- Abstract: Summary of the following sections and results
- Implementation: Description of the methodology used for implementing the proposed solutions (e.g. relevant classes and data structures, structure of the program, etc.).
- Experiments: Description of the experiments performed and obtained results. Results should be represented in a graphical form as well as discussed in the writing.
- Conclusions: Summary of the work and final considerations.
Similar sections can be merged if a single report is provided. However, the description of the implementation and experimental results should be well developed and discussed.