THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NETWORK RECOVERY AFTER AN AMBIGUOUS MASSIVE FAILURE

SEAMUS HAYES
SPRING 2016

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Computer Engineering
with honors in Computer Engineering

Reviewed and approved* by the following:

Tom La Porta
Professor of Computer Science and Engineering
Thesis Supervisor

Chita Das
Professor of Computer Science and Engineering
Honors Adviser

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

This thesis addresses the problem of finding the minimum cost set of repairs to perform on a destroyed communications network in order to support mission critical services. The work presented here is an expansion on the work done by N. Bartolini, S. Ciavarella, T. La Porta, and S. Silvestri in their paper *Network recovery after massive failures*. Their paper defines a polynomial time heuristic, called Iterative Split and Prune (ISP) that solves for the minimum cost set of repairs to be performed on a known disruption by recursively dividing the problem. We propose a new heuristic, *k*-Hop Iterative Split and Prune, which can solve for the minimum cost set of repairs when the exact structure of the network disruption is not known. We performed extensive simulations by varying the demand intensity, the number of demands, and the strength of the disruption. We find that the additions to the new *k*-Hop ISP enable it to perform well when the pattern of the destruction in partially unknown. The number of network elements that truly need repairing found by *k*-Hop ISP with partial knowledge is similar to the number of elements determined by ISP with full knowledge of the disruption.

TABLE OF CONTENTS

LIST OF FIGURES iii

ACKNOWLEDGEMENTS v

Chapter 1 Introduction 1

Chapter 2 Related Work 3

Chapter 3 Iterative Split and Prune 4

- Demand Based Centrality 4
- Split of the Demand 5
- Recovery of Nodes and Edges 6
- Pruning 6

Chapter 4 *k*-Hop Iterative Split and Prune 8

- Gray State 9
- Monitor Node 9
- Information Gain 9

Chapter 5 Experiments 14

Chapter 6 Conclusion 27

BIBLIOGRAPHY 28

LIST OF FIGURES

Figure 1. Equation for calculating the centrality of a node..5

Figure 2. A split of d_x units from the demand d_h across v_{bc}..5

Figure 3. Example of *k* units of demand being pruned from demand *d*.................................7

Figure 4. A demonstration of the information gain process on a sample graph with a small disruption 13

Figure 5. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with one demand with a demand intensity of three..15

Figure 6. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with one demand with a demand intensity of three...16

Figure 7. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with two demands with a demand intensity of three..17

Figure 8. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with two demands with a demand intensity of three...18

Figure 9. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with three demands with a demand intensity of three...19

Figure 10. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with three demands with a demand intensity of three.......................................20

Figure 11. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and k-Hop ISP heuristics on the largely disrupted BellCanada topology with one demand with a demand intensity of seven...21

Figure 12. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and k-Hop ISP heuristics on the largely disrupted BellCanada topology with one demand with a demand intensity of seven 22

Figure 13. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and k-Hop ISP heuristics on the slightly disrupted BellCanada topology with one demand with a demand intensity of three..23

Figure 14. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and *k*-Hop ISP heuristics on the slightly disrupted BellCanada topology with one demand with a demand intensity of three...24

Figure 15. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and k-Hop ISP heuristics on the slightly disrupted BellCanada topology with two demands with a demand intensity of three 25

Figure 16. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and k-Hop ISP heuristics on the slightly disrupted BellCanada topology with two demands with a demand intensity of three.......................................26

ACKNOWLEDGEMENTS

I would like to thank Professor La Porta for giving me the opportunity to work with him in his research and for his guidance throughout the process of completing my thesis. In addition, I would like to thank Professor Bartolini, Hana Khamfroush, Diman Tootaghaj, and Stefano Ciavarella for their invaluable assistance and teaching.

Chapter 1

Introduction

Communication networks serve an important role in relief efforts after a natural disaster or attack. However, communication networks, power networks, and emergency control networks are often critically damaged during these events [4]. The inability to communicate greatly impedes the effort of relief teams. This leads to unnecessary deaths and damage to property. Furthermore, as the tools used in disaster relief efforts become more and more dependent on communication networks, the impact of the network's destruction increases [2].

Hurricane Katrina provides great insight into the types of destruction caused by such disasters. According to the Federal Communication Commission (FCC), Katrina flooded much of the backbone conduit for line communication services as well as many of the central switching centers. In addition, Katrina brought down about 100 radio stations and 2000 cell towers, greatly degrading wireless communication capabilities. The loss of electrical power brought down communication stations that were not physically destroyed but were running on generators after their reserves of energy were drained [3].

The Federal Emergency Management Agency's (FEMA) Mobile Emergency Response Support (MERS) teams were designed to provide emergency communications, but were ineffective helping in the beginning stages of the relief effort [3]. It is necessary to develop some way to determine efficient repair strategies for communication networks.

The problem of satisfying the requirements of mission critical services is described in [1] as such:

We model the mission critical services as a demand graph. This graph defines a set of demand flows on the communication network, to which we refer to as supply network. We consider scenarios in which a major disruption of the supply network makes it unable to meet the capacity requirements of demand flows. Therefore, the flows must be accommodated by means of recovery actions. This model can be applied to any set of demand and supply networks.

In that same paper, the authors define a polynomial time heuristic, called Iterative Split and Prune (ISP), that attempts to solve for the minimum cost set of repairs when the structure of the disruption is known. We will propose a modification to ISP that performs well when the structure of the disruption is partially known, as it would most likely be in any real-use case. We call this new algorithm *k*-Hop ISP. We will test the role that knowledge of the destruction plays in ISP by varying the extent of *k*-Hops ISP's discovery phase. We will compare *k*-Hop ISP to the optimal solution and the solution found by the original version of ISP with full-knowledge of the destruction.

Chapter 2

Related Work

This thesis is an extension of work done in by Bartolini, et al. in [1]. In that paper the authors define the problem statement as a MILP and show that it is NP-hard. They propose the original ISP algorithm as introduced in chapter one and compare it to a number of greedy heuristics. They find it performs better than those heuristics and results in no demand loss.

Wang, el al. in [7] present a problem similar to our own. Instead of finding a set of repairs to restore a specific connectivity and capacity, they were interested in finding the optimal schedule of repairs to gain the most capacity across a source-demand pair as quickly as possible. The authors propose a heuristic that used MILP and LP sensitivity analysis to generate a shadow price for each potential piece of hardware to be repaired. Those components with the highest price are most likely to contribute to maximum the capacity gain.

Magnanti and Raghavan, in [6], evaluate the design of networks with specific connectivity constraints. They find that designing paths through a network to satisfy specific source-destination pair connectivity is an instance of the Steiner Forest problem. By adapting our thinking to consider network repair instead of network design, we can apply their flow-based formulations in our own algorithms.

Other works evaluate the recovery of electrical networks after a natural disaster [9, 10]. Other papers evaluate designing networks better able to weather a disaster and propose a way to measure a network's resilience [8, 11].

Chapter 3

Iterative Split and Prune

ISP relies on a metric called demand based centrality, which measures the importance of a node in the supply graph for routing the demand flows. The ISP algorithm splits the demand across the node with the highest centrality, forcing the flow to pass through that node. If the node is not working, it is repaired. After a split, ISP checks if the current supply network is able to satisfy any of the existing demand flows. If so, that demand is pruned, meaning the capacity needed by that demand is removed from the supply network. ISP terminates when there is no demand left to be routed.

Demand Based Centrality

This metric is used to determine which nodes to split demand across. This metric measures each individual node's ability to accommodate demand flows in the network.

To calculate the centrality of a node, let $P(i,j)$ be the set of acyclic paths in the demand graph connecting nodes i and j, both of which belong to the supply graph. Let $P^*(i,j)$ be the set of the shortest paths in $P(i,j)$. The length of the path is a function of both the number of edges in the path, and the status of the nodes and edges. Operational nodes and edges have a lower length cost than broken nodes or edges. Define the capacity of a path, $c(p)$, as the minimum capacity of all edges in a path, p. The centrality of each node v, $c_d(v)$, is the sum of the capacities $c(p)$ for each path $p \in P^*(i,j)|_v$ (meaning the paths which pass through v) divided by the sum of capacities c(p) for all $p \in P^*(i,j)$ multiplied by the demand. d_{ij}, of (i,j) for each $(i,j) \in E_H$, where E_H, in the set of demands.

$$c_d(v) \triangleq \sum_{(ij)\in E_H} \left(\frac{\sum_{p\in P^*_{ij}|_v} c(p)}{\sum_{p\in P^*_{ij}} c(p)} \cdot d_{ij} \right)$$

Figure 1. Equation for calculating the centrality of a node

Split of the Demand

The node with the highest demand based centrality is chosen for a split. This node is called the *best candidate*. In the event that the set of nodes with the highest centrality is larger than one, any node in the set can be chosen as best candidate.

Let a demand of d_H units exist between the pair of nodes, (s_H, t_H). To split the demand between (s_H, t_H) across the best candidate, v_{BC}, remove d_X units of demand from demand pair (s_H, t_H) and add that demand into two new demand pairs (s_H, v_{BC}) and (v_{BC}, t_H). The value d_X is chosen such that $d_X \leq d_H$ and $d_X \leq c(p)$ for $p \in P^*(s_H,t_H)|_{vbc}$.

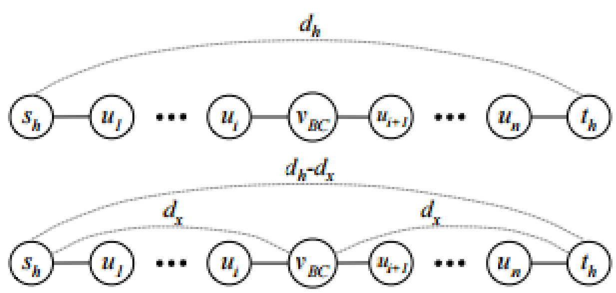


Figure 2. A split of d_x units from the demand d_h across v_{bc}

Recovery of Nodes and Edges

The ISP algorithm works by progressively recovering nodes and edges. When these repairs are reflected in the supply graph, the problem statement is greatly changed and reduced. Eventually, these repairs give us a supply graph which is capable of supplying all demands.

If a broken node is selected as *best candidate*, it is repaired. Furthermore, if there is a edge that directly connects two endpoints of a demand which cannot be satisfied by the supply graph at the current iteration, that edge is repaired.

Pruning

Pruning is the act of removing an amount of demand in order to simplify the problem. When all demands are pruned, ISP terminates with a valid solution. For a demand to be able to be pruned, there must exist some working path between the endpoints of the demand in the supply graph. If the capacity of that path is equal or greater than the flow to be pruned, the entire demand can be removed from the problem. However, if the capacity of the path is less than the demand of the flow, only that fraction of the flow can be pruned and the rest of the demand must be satisfied through some other path.

Just because a demand is able to be pruned, however, does not mean it will be. Pruning, like splitting, implies a routing decision. Therefore, it is possible that pruning may lead to an infeasible solution. In order to ensure this does not occur, ISP defines a set of nodes called a bubble which, when pruned, ensure feasible solutions. A path from (s_h, t_h) is a bubble if it contains only nodes that cannot be reached by any other demand endpoint without traversing s_h or t_h.

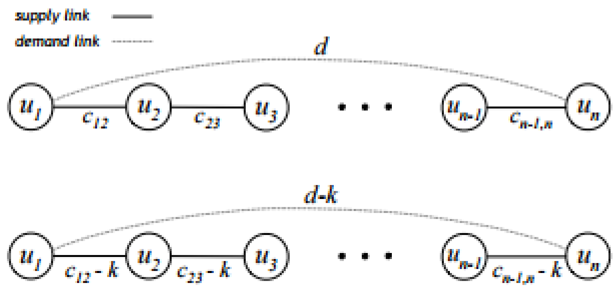


Figure 3. Example of *k* units of demand being pruned from demand *d*

Chapter 4

k-Hop Iterative Split and Prune

The original version of ISP operates under the assumption that the complete pattern of the disruption is known. The main benefit of *k*-Hop ISP is that it performs well when the pattern of the disruption is partially unknown and cannot be quickly determined. In order to model the unknown disruption, *k*-Hop ISP includes a gray state to represent nodes whose status in unknown. *k*-Hop ISP makes up for the lack of knowledge by including a new information gain phase in which the status of the network is measured by means of properly placed monitoring nodes. Repairing a node or edge may restore connectivity to a previously unreachable part of a network so that the exact state of gray nodes may be learned, or it may increase the amount of flow able to pass to and from a pair of nodes. The information gain phase progressively determines the structure of the disruption of a network as components are repaired and monitors are installed. The information gained is used in the routing and repair decision in the continued execution of the algorithm.

Algorithm: K-Hop Iterative Split and Prune (*k*-Hop ISP)

Input: Supply Graph G, demand graph H, broken nodes V_B and broken edges E_B

while *routability test fails* **do**
  Information Gain on G and H;
  while *pruning condition* **do**
    Prune demands satisfying pruning condition;
    Update G and H;
    Information Gain on G and H;
  if *there are repairable links* **then**
    Repair;
    Update G and E_B;
    Information Gain on G and H;
  else
    Find *best candidate* v_{BC} for split;
    Find best demand d to split on v_{BC} ;
    Calculate the maximum splittable amount d_x;

Split amount d_x of demand d on v_{BC} ;
Update G, H, V_B;
Information Gain on G and H;

Gray State

In the original ISP, nodes and edges can have one of three possible statuses: *on*, *repaired*, and *destroyed*. The *on* status means that the node or edge is in working order. The *repaired* status means that the node or edge was destroyed, but has been selected for repaired. The *destroyed* status means that node or edge is not working. In *k*-Hop ISP, the *gray* status is for nodes or edges that cannot be classified as *on* or *destroyed*. This can happen when a node or edge is disconnected from or not within the information gain distance of any monitoring node.

Monitor Node

Nodes that are part of a demand pair or that have been repaired are monitoring nodes. This means, *k*-Hop ISP is able to monitor traffic through these nodes and use them to test connectivity to other network hardware through pings. Monitoring nodes are used as base nodes for discovering the structure of the network disruption in the information gain phase.

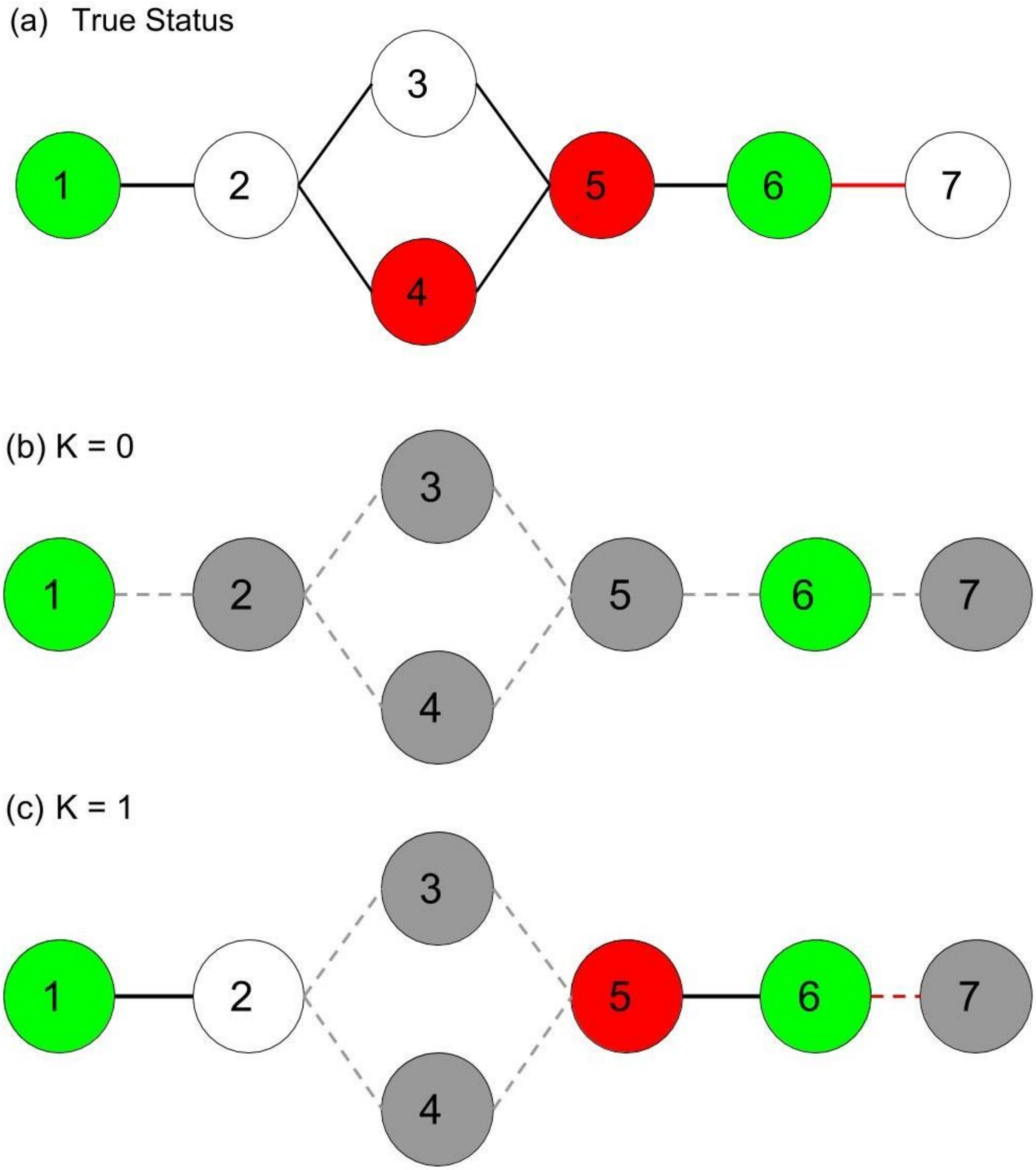
Information Gain

In the information gain phase, *k*-Hop ISP probes the network in order to discover knowledge regarding the structure of the disruption. This knowledge is used to convert nodes with the *gray* status to have a status of either *on* or *destroyed*, allowing *k*-Hop ISP to more accurately make repair and routing decisions. The information gain process is run at the beginning of every iteration of the *k*-Hop ISP algorithm as well as when any node or edge is selected for repair. A monitor is installed on all repaired nodes, allowing information about the network to be gained.

The *k* in *k*-Hop ISP is a variable that represents the maximum distance the information gain algorithm can search through the network. Distance is defined by number of edges traversed. Therefore, if *k* is equal to 0, nothing can ever be discovered. If *k* is set to one, nodes and edges one hop away from all monitor nodes can be discovered. If *k* is set to two, nodes and edges two hops away from any monitor node can be discovered. *k* can be any natural number.

Discovery originates at every monitor node. The status of edges attached to a monitor node can be determined through the presence of carrier signals. If a monitor node detects a carrier signal on an attached line, it can be sure that edge is in working order. If no signal is detected, the edge must be broken. For all the working edges, the status of the attached node can be determined through a simple ping. If it was the case that the edge was broken, the status of the attached node cannot be determined unless it is reachable by some other path. Past its attached edges, the monitor nodes cannot detect carrier signals. Therefore, nodes and edges with greater than one-hop distance from a monitor node can only be determined as working if the edge and attached node are both working. If either is failed, it is impossible to tell if the edge or the node or both is what is causing the failure. This means both the node and edge must remain *gray*.

Figure 4 below, shows how the discovery algorithm works with different levels of discovery (values of *k*). The green nodes represent monitoring nodes, the gray nodes are those whose status is unknown, red nodes are broken, and white nodes are operational. As with nodes, gray edges are those whose status is unknown, red edges are broken, and black edges are working. The graph in (a) shows the true status of the nodes and edges in the network. In (b), when $k = 0$, nothing can be discovered, so all nodes and edges excluding the monitoring nodes remain gray. In (c), when $k = 1$, the status of all the edges connected to the monitoring nodes can be determined with certainty. Because the edges (1,2) and (5,6) are working, the status of nodes 2 and 5 can be determined through a simple ping. The edge (6,7) is broken, meaning node 7 cannot be pinged and its status cannot be determined. No more information can be gained from node 6 because the propagation of its discovery is blocked by the broken node 5 and broken edge (6,7). The graph in (d) shows what can be discovered at $k = 2$. Referring to the true status graph, we see node 3 is working and node 4 is broken. Both edges (2,3) and (2,4) are working. Because edge (2,3) is working and the node 3 is working and within 2 hops of monitor node 1, the discovery process is able to ping node 3 and determine the edge (2,3) and the node 3 are both operational. Node 4, however, is broken and cannot be pinged. From the point of view of the monitoring node, it is impossible to determine which components are responsible. Both edge (2,4) and node 4 remain gray. The graph for when $k = 3$, shown in (e) is identical to the graph (d). Because node 5 is destroyed, no pings from the monitor node 1 can reach it. This means we are unable to determine the status of edges (3,5) and (4,5). In this case, the graph will remain the same for all values of k larger than or equal to two.


(a) True Status
3
1
2
5
6
7
4
(b) K = 0
3
1
2
5
6
7
4
(c) K = 1
3
1
2
5
6
7
4

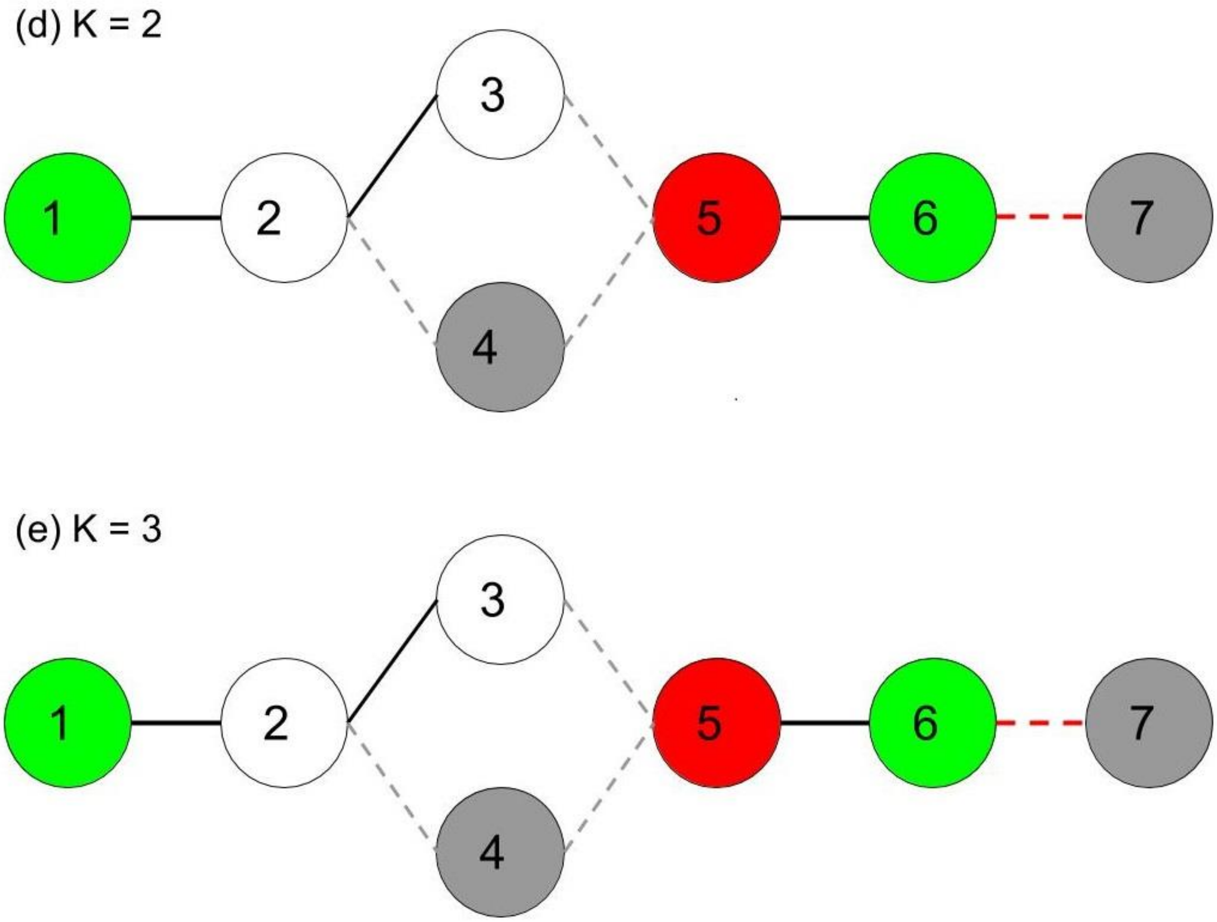


Figure 4. A demonstration of the information gain process on a sample graph with a small disruption

Chapter 5

Experiments

We ran a variety of simulations to compare *k*-HOP ISP to the original ISP and the optimal solution. To perform these simulations, we used Bell Canada's real network topology, found in the Internet Topology Zoo [5]. We varied the number of demand pairs, the size of the demand, and the scale of the disruption. We measured the number of nodes and edges selected for repair by the original ISP with full knowledge of the disruption, by *k*-Hop ISP with varying degrees of *k*, and by the optimal solution. We made a distinction between necessary and unnecessary repairs. The charts labeled 'REAL REPAIRS' only count those network components which needed actual repairs (Repair of red node/edge or a gray node/edge that was found to broken)

Figure 5 through *Figure 10* show the effect of increasing the number of demand pairs from one to three. Each of these demand pairs has three unites of demand to be satisfied. As shown in the charts, ISP performs nearly as well as the optimal solution. When $k = 0$, meaning there is no information gain, *k*-Hop ISP requires significantly more repairs than the optimal solution, ISP's solutions, or *k*-Hop ISP solutions with higher degrees of information gain. To satisfy one pair of demand requiring three units of flow, ISP required about six repairs on average. *k*-Hop ISP with $k = 0$ required nearly 16 repairs, on average. When *k* was higher, the average number of repairs dropped to 11. *k*-Hop ISP and ISP make a similar number of real repairs, but *k*-Hop IPS's lack of information caused it to select working node for unneeded repairs.

Comparing *Figure 5* and *Figure 6* with *Figure 11* and *Figure 12* shows the effect of varying the intensity of the demand. Similarly to varying the number of pairs, the ISP algorithm performs near optimally. *k*-Hop ISP performs closely to ISP when real repairs are considered, but makes many unnecessary repairs, especially so when $k = 0$.

Comparing *Figure 6* through *Figure 8* with *Figure 13* through *Figure 16* shows how the intensity of the disruption affects the performance of the algorithms. In the small disruption, the information gain phase is better able to permeate the network, so more stark improvements are seen by increasing *k* than in the larger disruptions.

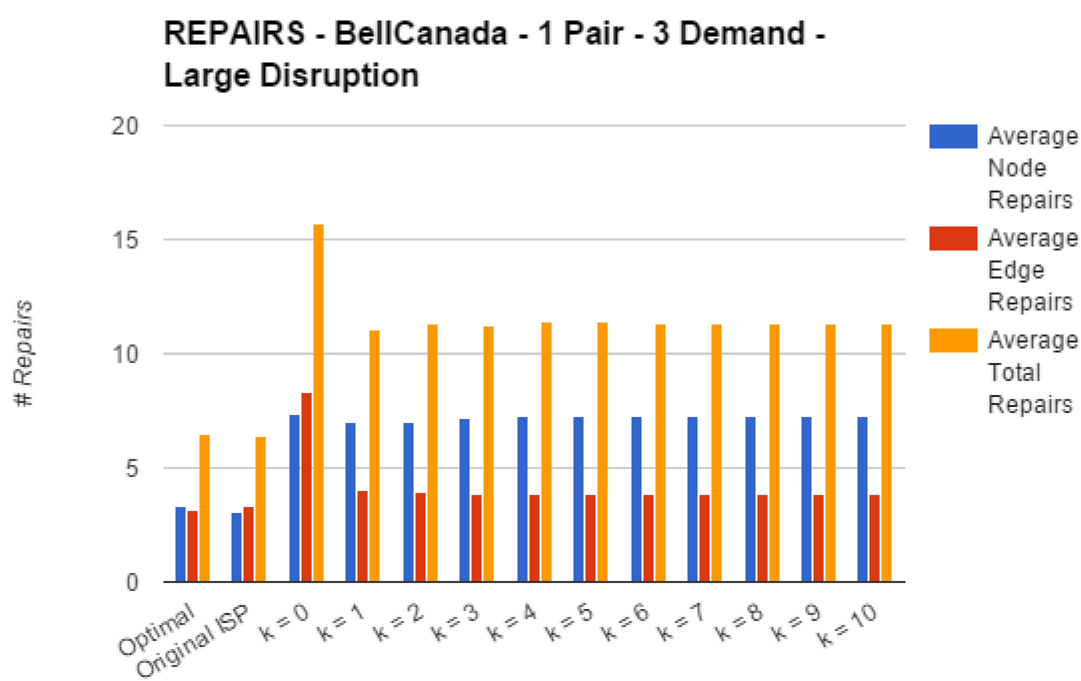


Figure 5. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with one demand with a demand intensity of three

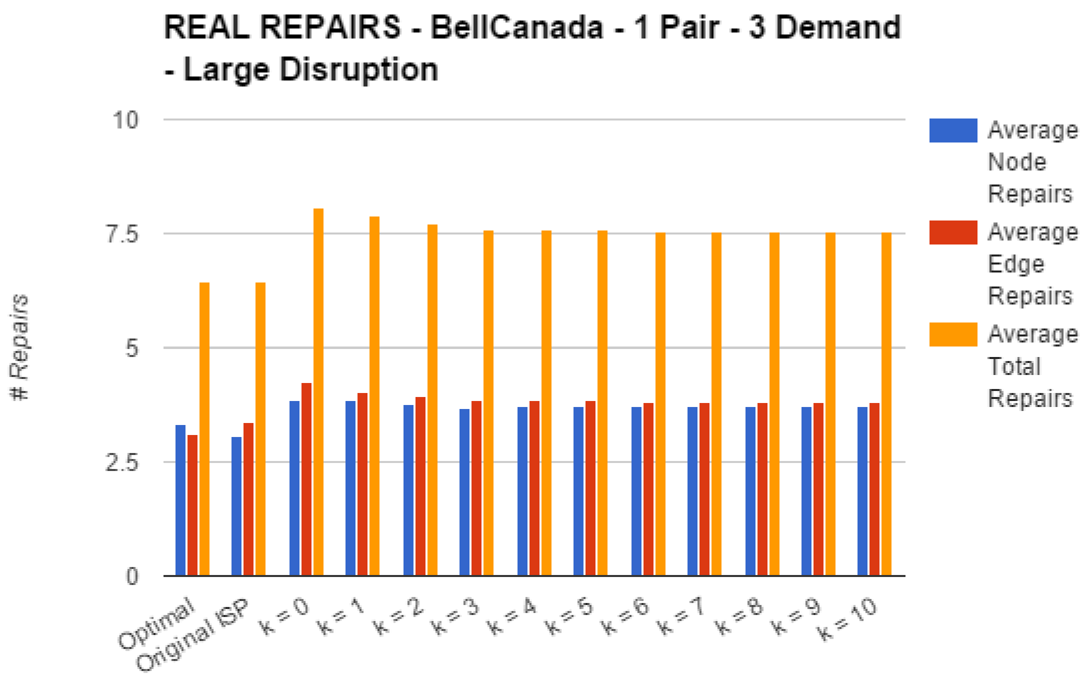


Figure 6. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with one demand with a demand intensity of three

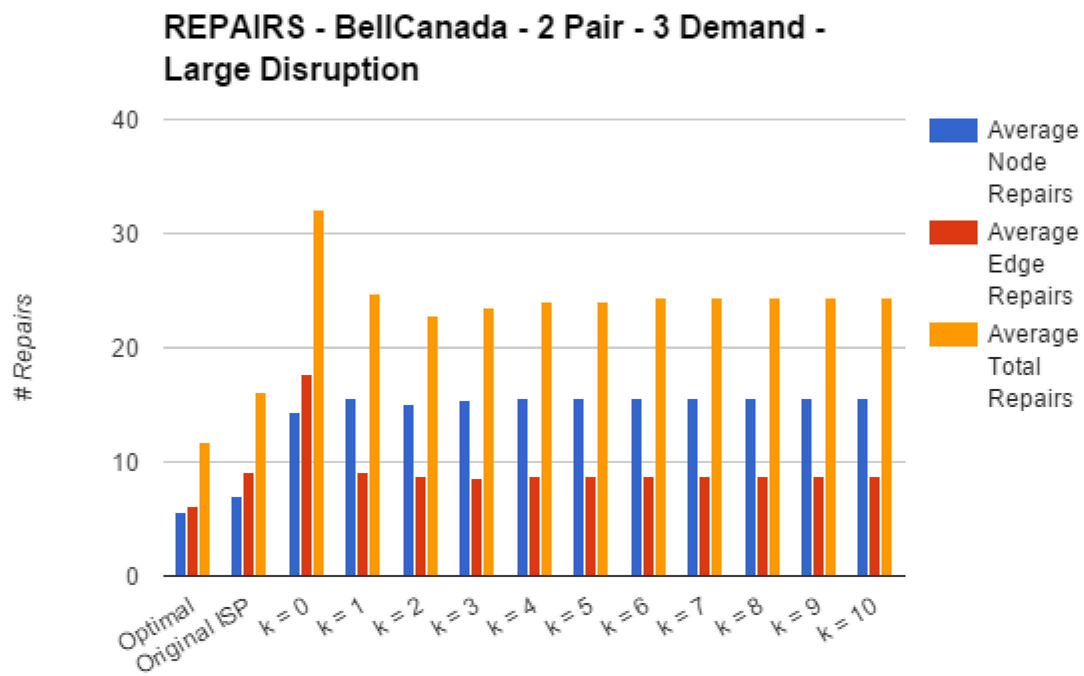


Figure 7. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with two demands with a demand intensity of three

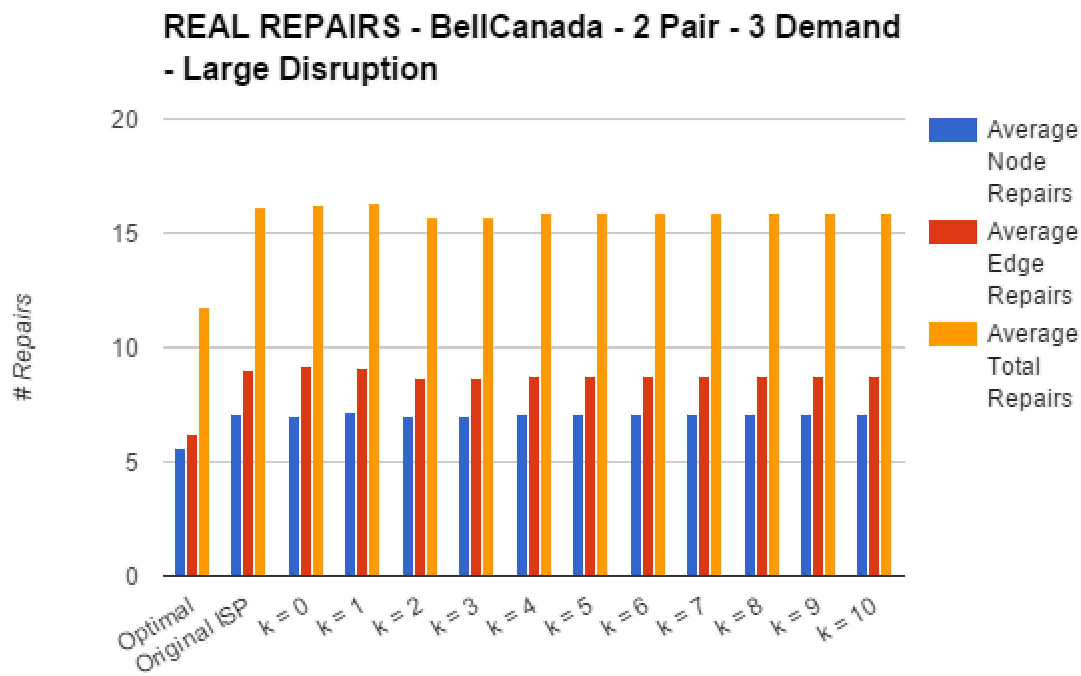


Figure 8. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with two demands with a demand intensity of three.

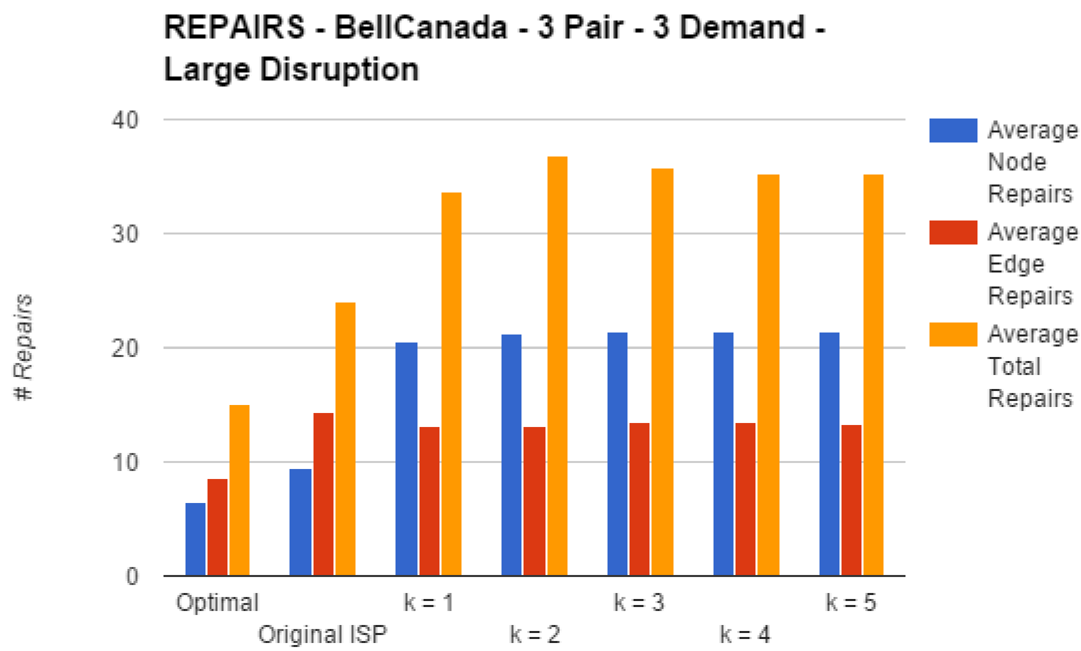


Figure 9. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with three demands with a demand intensity of three.

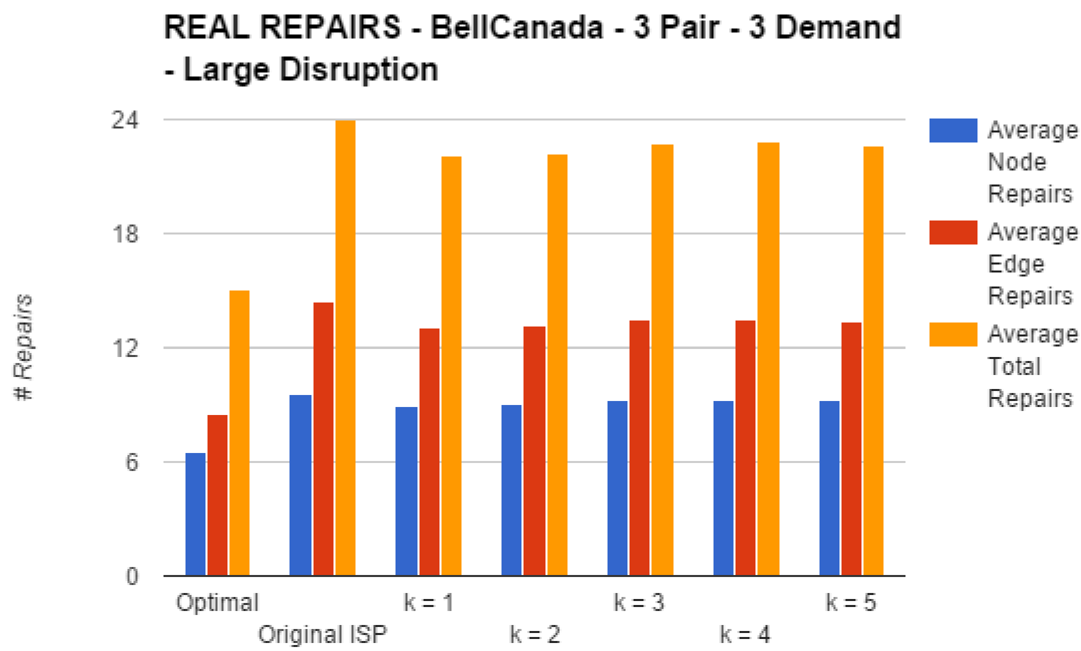


Figure 10. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and *k*-Hop ISP heuristics on the largely disrupted BellCanada topology with three demands with a demand intensity of three

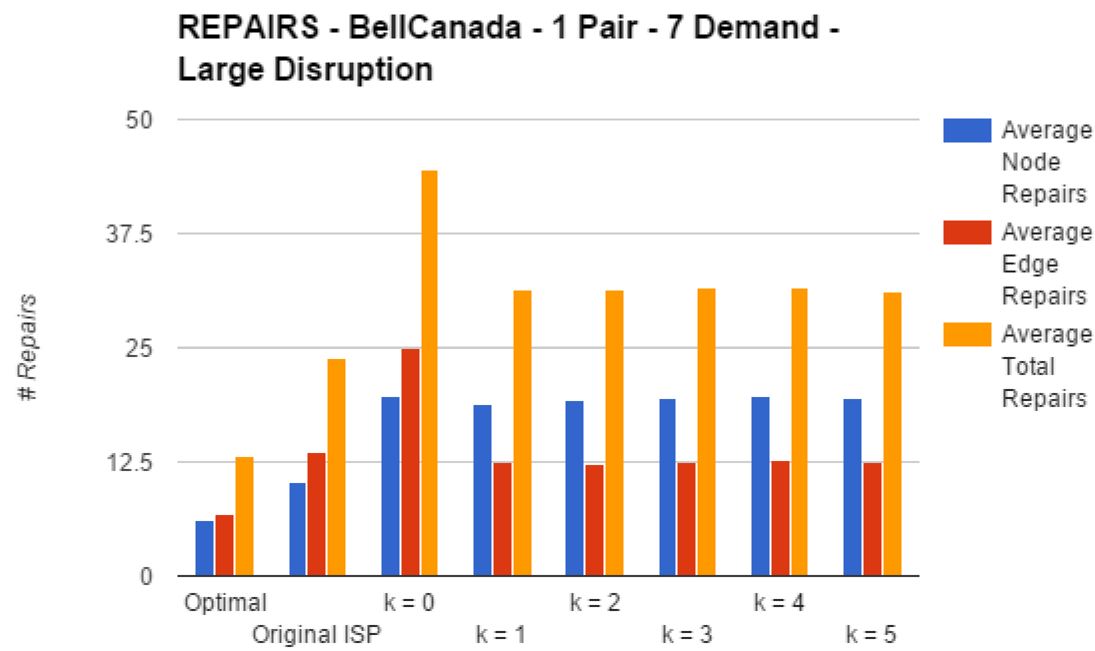


Figure 11. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and k-Hop ISP heuristics on the largely disrupted BellCanada topology with one demand with a demand intensity of seven

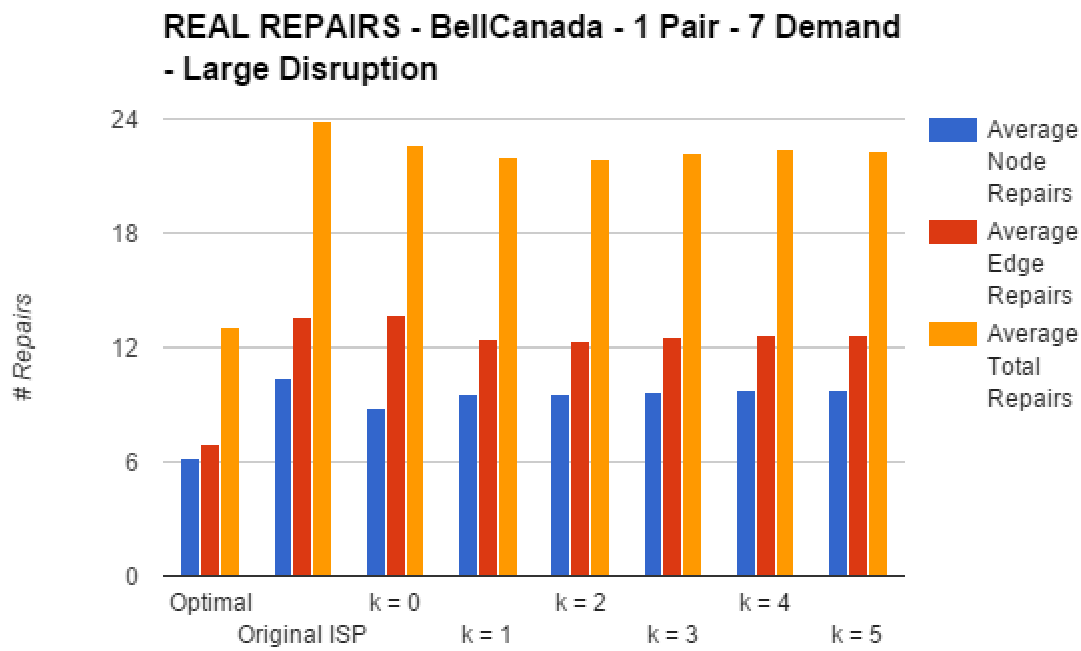


Figure 12. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and k-Hop ISP heuristics on the largely disrupted BellCanada topology with one demand with a demand intensity of seven

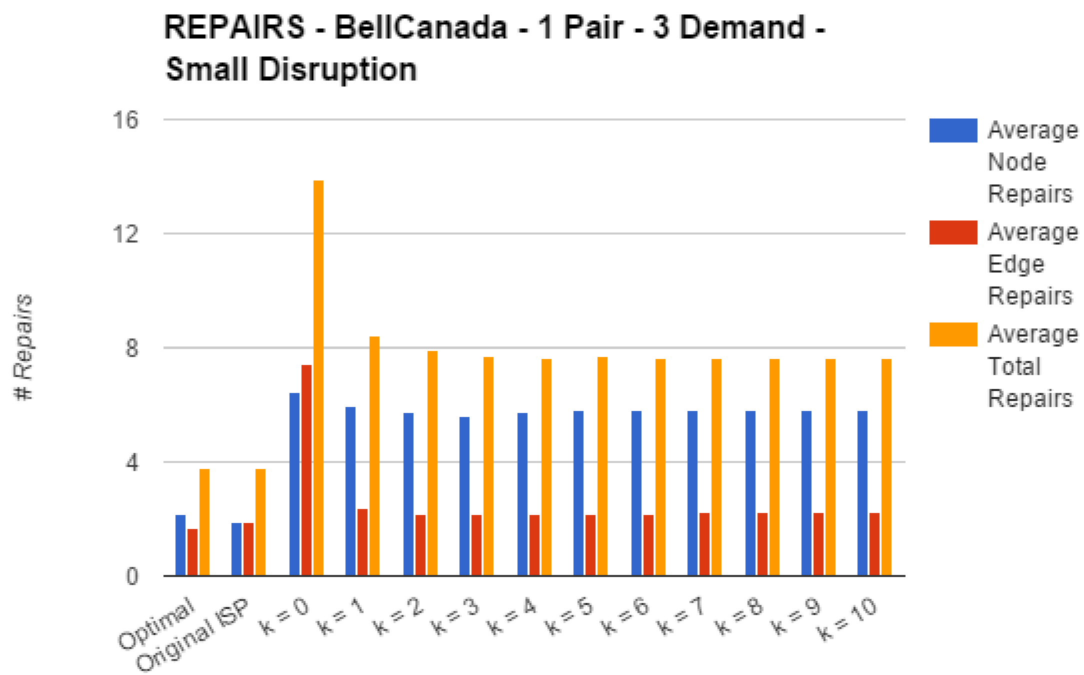


Figure 13. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and k-Hop ISP heuristics on the slightly disrupted BellCanada topology with one demand with a demand intensity of three

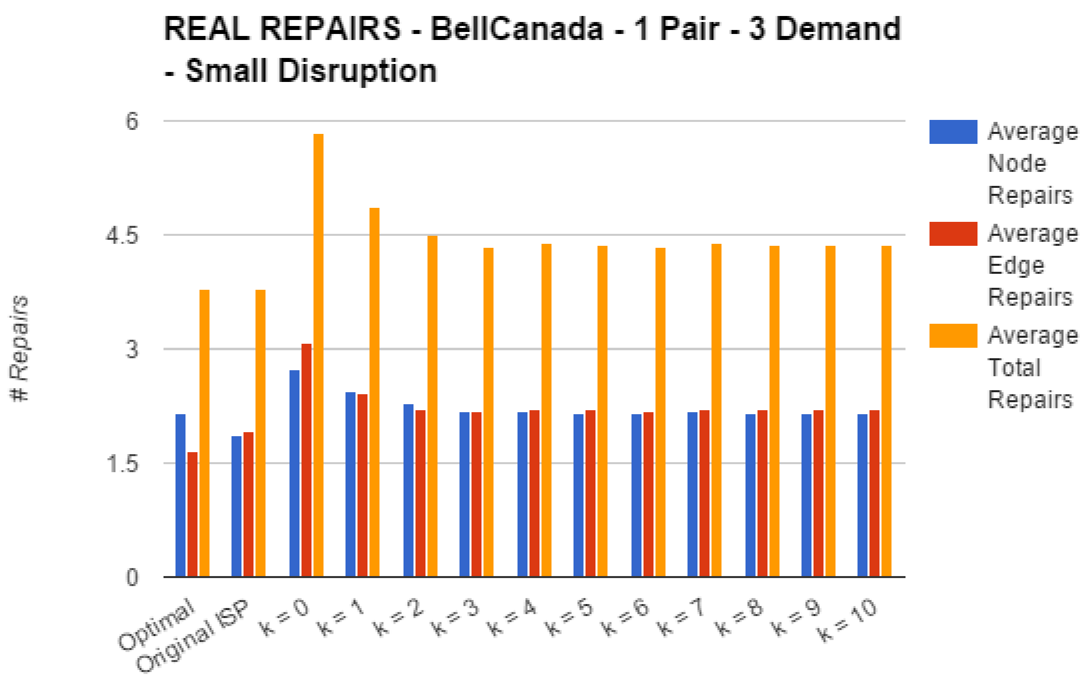


Figure 14. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and *k*-Hop ISP heuristics on the slightly disrupted BellCanada topology with one demand with a demand intensity of three

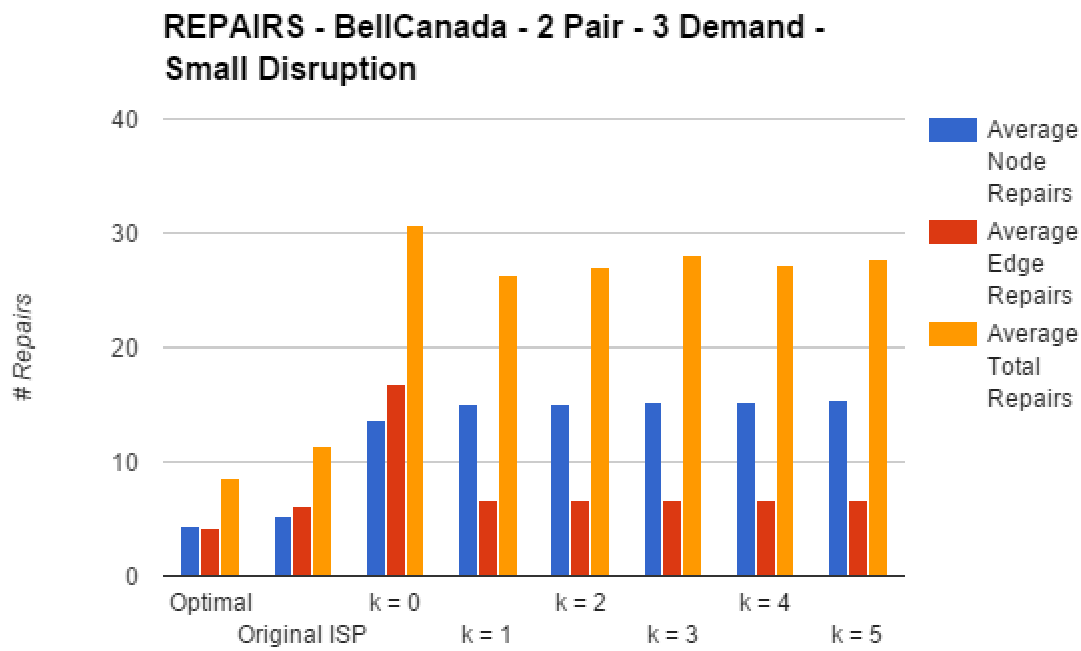


Figure 15. Chart comparing the number of nodes and edges selected for repair by the optimal, original ISP, and k-Hop ISP heuristics on the slightly disrupted BellCanada topology with two demands with a demand intensity of three

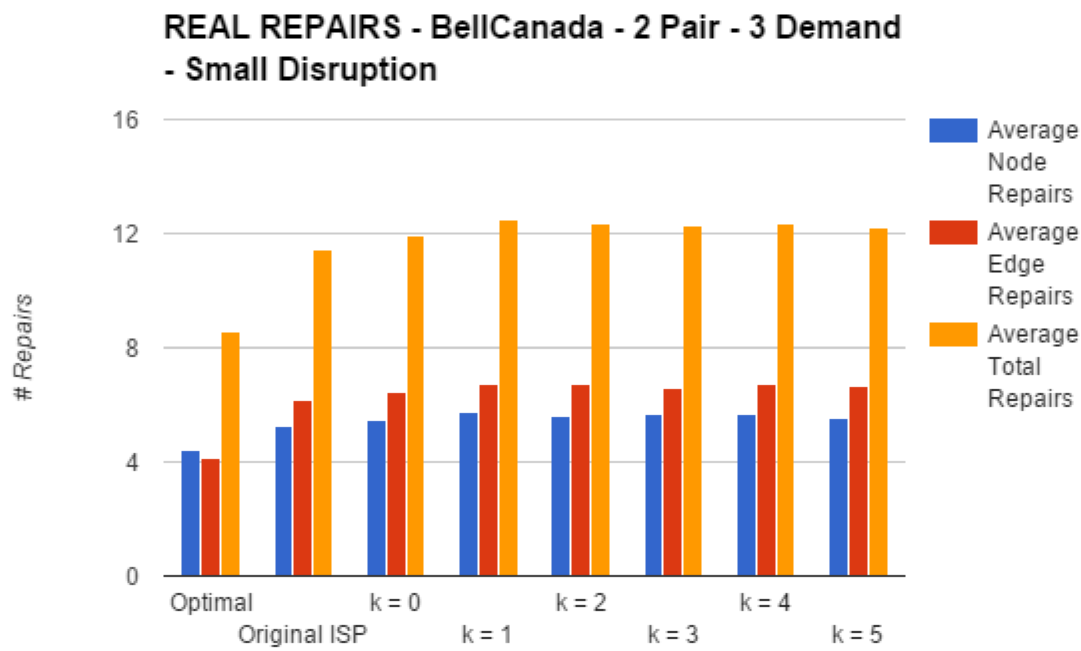


Figure 16. Chart comparing the number of nodes and edges actually needing repair selected by the optimal, original ISP, and k-Hop ISP heuristics on the slightly disrupted BellCanada topology with two demands with a demand intensity of three

Chapter 6

Conclusion

In this thesis, we consider the same problem of recovery of a communication network after a large disruption as presented in [1]. We propose an extension to ISP, called *k*-Hop ISP, to enable it to perform well when the pattern of the destruction in partially unknown. The *k*-Hop ISP features a new information gain stage needed to determine that status of network components for routing and repair decisions.

Experiments on a real network topology show increasing the scale of the information gain phase in *k*-Hop ISP leads to fewer repairs to satisfy the required demand. Increasing *k*, the degree of information gain, improves the algorithm's performance. The most striking improvement is seen between $k = 0$ and $k = 1$. The improvements cease after a certain value of *k* when there is no information left to be gained. However, this value of *k* is dependent on the specific problem instance.

BIBLIOGRAPHY

[1] N. Bartolini, S. Ciavarella, T. F. La Porta, S. Silvestri, “Network Recovery After Massive Failures,” IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2016), to appear.

[2] A. Townsend, M. Moss. “TELECOMMUNICATIONS INFRASTRUCTURE IN DISASTERS: Preparing Cities for Crisis Communications”. Center for Catastrophe Preparedness and Response & Robert F. Wagner Graduate School of Public Service New York University. 2005

[3] R. Miller.” Hurricane Katrina: Communications & Infrastructure Impacts”. National Defense University.

[4] W. M. Leavitt and J. J. Kiefer. “Infrastructure interdependency and the creation of a normal disaster the case of hurricane Katrina and the city of New Orleans”. Public works management & policy, 10(4):306–314, 2006.

[5] “The internet topology zoo”. http://www.topologyzoo.org/. Last accessed on May 2015.

[6] T. L. Magnanti and S. Raghavan. “Strong formulations for network design problems with connectivity requirements”. Networks, 45(2), 2005

[7] J. Wang, C. Qiao, and H. Yu. “On progressive network recovery after a major disruption”. In IEEE INFOCOM, 2011.

[8] A. Kwasinski, "Technology planning for electric power supply in critical events considering a bulk grid, backup power plants, and micro-grids", *IEEE Syst. J.*, vol. 4, no. 2, pp. 167-178, 2010

[9] A. Arab, A. Khodaei, Z. Han, and S. Khator “Proactive recovery of electric power assets for resiliency enhancement”. Access, IEEE, 3, 2015.

[10] B. Ball, "Rebuilding electrical infrastructure along the gulf coast: A case study", *Nat. Acad. Eng.*, vol. 36, no. 1, pp. 21, 2006

[11] P. J. Maliszewski and C. Perrings, "Factors in the resilience of electrical power distribution infrastructures", *Appl. Geogr.*, vol. 32, no. 2, pp. 668-679, 2012

ACADEMIC VITA

Academic Vita of Seamus Hayes
sih5349@psu.edu

Education
Major(s) and Minor(s): Computer Engineering

Thesis Title: Network Recovery after an Ambiguous Massive Failure
Thesis Supervisor: Professor Tom La Porta

Work Experience
Ruckus Wireless
Summer 2015
Software Engineering Intern

Cisco Systems
Summer 2015
Software Engineering Intern