Enhancing Network Robustness via Shielding

by

Jianan Zhang

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

ARCHIVES
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
OCT 0 2 2014
LIBRARIES

© Massachusetts Institute of Technology 2014. All rights reserved.

Signature redacted

Author .....
Department of Aeronautics and Astronautics
August 21, 2014

Signature redacted

Certified by .....
Eytan Modiano
Professor of Aeronautics and Astronautics
Thesis Supervisor

Signature redacted

Accepted by .....
Paulo C. Lozano
Chairman, Graduate Program Committee

Enhancing Network Robustness via Shielding

by

Jianan Zhang

Submitted to the Department of Aeronautics and Astronautics
on August 21, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Shielding critical links enhances network robustness and provides a new way of designing robust networks. We first consider shielding critical links to guarantee network connectivity after any failure under geographical and general failure models. We develop a mixed integer linear program (MILP) to obtain the minimum cost shielding to guarantee the connectivity of a single source-destination (SD) pair under a general failure model, and exploit geographical properties to decompose the shielding problem under a geographical failure model. We extend our MILP formulation to guarantee the connectivity of the entire network, and use Benders decomposition to significantly reduce the running time by exploiting its partial separable structure. We extend the algorithms to guarantee partial network connectivity, and observe that significantly less shielding is required, especially when the failure region is small.

To mitigate the effect of random link failures on network connectivity, we consider increasing the effective min-cut of the network by shielding, where shielded links cannot be contained in effective cuts. For a single SD pair, we develop an efficient algorithm to increase the effective min-cut by one, and develop a MILP with a small number of constraints to increase the effective min-cut by an arbitrary value. Then we extend the MILP to obtain the optimal shielding to increase the effective min-cut for the entire network, which can be used to solve realistic size problems.

Finally, we consider shielding critical nodes in random graphs. We demonstrate the importance of high degree nodes in random graphs constructed under the configuration model. The occupancy of higher degree nodes leads to a larger size of the giant component. Moreover, shielding a small fraction of nodes in power law random graphs guarantees the existence of a giant component if the exponent is less than three.

Thesis Supervisor: Eytan Modiano
Title: Professor of Aeronautics and Astronautics

Acknowledgments

First and foremost, I would like to express my gratitude to my advisor Professor Eytan Modiano. His guidance and support in the past two years has been a driving force for me, especially when I could not move any further in research. He is an extremely good listener, with huge patience to discuss research problems. His careful reading of the thesis and comments improved my writing and the quality of this thesis.

Next, I would like to thank my labmates. In particular, I would like to thank Marzieh Parandehgheibi for academic discussions, and Anurag Rai, Abhishek Sinha, and Kyu Soeb Kim for sharing interesting conversations, which makes late hours in the lab more colorful.

I also would like to thank my girlfriend Yi Wan for being with me and sharing sweet memories. Many thanks for her understandings and encouragements in my hard time of getting stuck in research.

Moreover, I would like to express my thanks to my parents for their eternal love, support and caring. They are always standing behind me to give me the freedom to explore while provide shelter when I want to rest. I could not have come this far without their love.

Last, this research was supported in part by Y. T. Li Fellowship and NSF grant CNS-1017800.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Communication networks are subject to natural disasters and attacks, such as hurricanes, earthquakes, and electromagnetic pulse attacks [21, 9]. Network failures may result in tremendous financial loss and hinder effective recovery to the affected regions. Therefore, it is important for network designers to guarantee that the network can withstand failures that may result from disasters or attacks.

Several metrics measure the performance of the network. The most fundamental requirement is connectivity, without which it is impossible to support any application that requires communication through the network. Another metric, important for quality of service guarantee, is the maximum amount of traffic carried by the network. In case of network failures, one cannot expect the network to support the same amount of traffic as before the failure. However, low priority applications such as online games and movies, can be throttled to give higher priority to critical applications in case of network failures. In this thesis, we focus on guaranteeing network connectivity, and assume that networks are able to use limited resources to support critical applications using service differentiation.

Previous research considers geographical failures [36, 1] and general failures [10, 27] to assess the robustness of the network. Geographical failure models capture the effects of natural disasters and physical attacks; e.g., all links in the failure region

are destroyed. Under the general failure model, an arbitrary set of links may fail, i.e., each failure may affect a specified set of links, whose number and location is determined by the nature of the failure.

A common approach to design robust networks is through redundancy and backup routes (see [32, 33] for a survey of protection techniques for optical networks). Backup routes guarantee connectivity in case of primary link failures, and most works use mathematical programming to obtain the optimal backup provisioning [10, 40]. An alternative approach, which we consider in this thesis, is shielding critical links. Shielded network infrastructure can survive disasters and attacks. Previous research suggests strengthening cables to resist physical attacks [34], and upgrading or covering vulnerable components to resist electromagnetic pulse attacks [8].

Due to the cost of shielding, it may not be economical to shield the entire network. Instead, critical parts of the network can be identified and shielded to guarantee network robustness. Previous work identifies critical parts to shield in order to achieve certain network performances in various applications [20, 31, 43, 19, 16, 12]. The authors in [20] design optimal topologies, given different levels of shielding cost, link construction cost, and utility of network connectivity, under the assumption of uniform costs for all links. In [31], the authors formulate a road network retrofit problem, and use a two stage stochastic programming approach to decide which roads to retrofit to minimize the average performance loss incurred by a disaster. Fortifying facilities to minimize the transportation cost and shortest path is also considered in [16, 12, 43].

In this thesis, we aim to design robust networks by shielding critical parts of the network. We determine the minimum cost shielding to guarantee that the network is connected after any failure under geographical and general failure models. In addition, we identify important links to shield, in order to increase the number of link failures that the network can tolerate. Finally, we consider shielding in random graphs and identify critical infrastructure based on local information of the graph, since globally optimizing in random graph is intractable using the traditional optimization approach.

1.2 Contributions

In Chapter 2, we aim to design robust networks by shielding critical parts of the network. We determine the minimum cost shielding to guarantee that the network remains connected after a failure, using both geographical and general failure models. First, we consider a single source-destination (SD) pair in the network and determine the minimum cost shielding to guarantee its connectivity. We develop a mixed integer linear program (MILP) to formulate the shielding problem in the case of general failures, and identify properties of optimal shielding in the case of geographical failures to decompose the shielding problem to multiple subproblems, each of which determines the optimal shielding for a disjoint set of links. Then, we extend the MILP formulation to consider guaranteeing the connectivity of the entire network under the general failure model. By identifying the partial separable structures of the MILP, we apply the Benders decomposition technique [5] to reduce the running time and solve network shielding problems of realistic size. A heuristic based on simulated annealing further reduces the running time significantly while achieving good results. Moreover, we observe that shielding cost can be significantly reduced if the connectivity requirement is slightly relaxed.

Chapter 3 considers shielding network links to enhance network robustness under the random failure model. Eliminating small cuts in a network is key in improving network reliability especially when the failure probability is small. The shielded links are considered to be reliable and cannot be contained in effective cuts. Thus cuts that contain the shielded links are eliminated. We develop an efficient algorithm to optimally eliminate all the min cuts for a single SD pair, and develop a MILP to eliminate all the small cuts whose size is smaller than a given threshold using the minimum cost. Then we consider eliminating the min cuts of the entire network and develop an efficient algorithm to provide sufficient shielding and obtain an upper bound on the optimal shielding cost. Finally we extend the MILP to obtain the optimal shielding to eliminate network cuts whose sizes are below certain threshold.

Chapter 4 considers shielding critical nodes in random graphs constructed under

the configuration model. We use the giant component size as a measure for graph connectivity. It is intractable to use traditional optimization techniques to determine a given number of critical nodes whose removal minimizes the size of the giant component since the number of nodes approaches infinity in the asymptotic case. Instead, we utilize degree information and theoretically justify in configuration model that the occupancy of higher degree nodes results in a larger giant component. In a power law random graph which has exponent less than three, shielding a small fraction of nodes guarantees the existence of a giant component in the asymptotic case.

Chapter 2

Shielding to guarantee network connectivity

2.1 Failure models and network shielding

Geographical failure models can be used to model real world disasters and attacks [36, 1, 14]. In this chapter, we consider the disk failure model, which captures the effect of electromagnetic pulse attacks. A disk failure with a given radius may occur anywhere in the network, and all the links in the disk region are affected. Multiple failure regions can be represented by one failure region which dominates them, where one failure dominates another failure if and only if it affects all the links affected by the other failure. The number of dominating failures is polynomial in the number of links and can be efficiently obtained by computational geometry techniques [36, 1].

In addition, we consider a general failure model that represents the failures in shared risk groups [10, 39]. Instead of being limited to be within a geographical region, the set of failed links can be arbitrary, possibly restricted by the nature of the attack or disaster. Under the general failure model, the possible failures and links affected by each failure are described explicitly.

We aim to shield links by using the minimum cost to guarantee network connectivity after any single failure event. For simplicity we assume that shielded links do not fail. We first consider guaranteeing the connectivity of a single SD pair, and later

extend to the connectivity of the entire network. Finally we relax the connectivity requirement for the entire network to allow for partial connectivity, which requires much less shielding cost.

2.2 Connectivity of a single SD pair

We start by considering the shielding problem in order to guarantee the connectivity of a single SD pair. It suffices to shield links to guarantee that a path will exist after any failure event. Clearly, if only one link can fail at a time, the links in the minimum-cut-equals-1-set need to be shielded. The minimum-cut-equals-1-set is the set of links among which any single link failure disconnects the SD pair. Any other single link failure will not disconnect the SD pair and need not be shielded. However, if a failure event affects several links and disconnects the SD pair, not all the affected links need to be shielded in order to guarantee a path after the failure event. We aim to determine the links that need to be shielded with minimum cost to guarantee the connectivity of the SD pair after any failure under both the general and geographical failure models.

2.2.1 Shielding under the general failure model

Under the general failure model a failure is specified by the set of failed links. The network is represented by a graph $G = (V, E)$. Each failure z affects a set of links $E^{(z)}$. The objective is to shield a set of links E^* with minimum shielding cost, to guarantee that s and d are connected through $G = (V, E'^{(z)})$ for all z, where $E'^{(z)} = (E \setminus E^{(z)}) \cup E^*$.

The optimal shielding problem under the general failure model can be modeled by MILP. Let $t_{ij}^{(z)}$ indicate whether or not failure z affects link (i, j). Failure z affects a set of links $E^{(z)} = \{(i, j) | t_{ij}^{(z)} = 1\}$. Each link (i, j) has shielding cost c_{ij}. Both $t_{ij}^{(z)}$ and c_{ij} are problem parameters. The decision variables are $x_{ij}^{(z)}$ and h_{ij}, which represent the amount of flow carried on link (i, j) after failure z and whether or not to shield link (i, j), respectively. The set of shielded links is $E^* = \{(i, j) | h_{ij} = 1\}$.

Since the links are undirected, (i,j) is the same link as (j,i), in which case $c_{ij} = c_{ji}$ and $h_{ij} = h_{ji}$. The minimum shielding cost to resist any possible failure is given by the following MILP.

$$\min \quad \sum_{(i,j)\in E} c_{ij}h_{ij}/2$$

$$\text{s.t.} \quad \sum_{\{j|(i,j)\in E\}} x_{ij}^{(z)} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(z)} = \begin{cases} 1, \text{ if } i = s \\ -1, \text{ if } i = d \\ 0, \text{ otherwise} \end{cases} \quad \forall z \tag{2.1}$$

$$x_{ij}^{(z)} - h_{ij} \leq 1 - t_{ij}^{(z)} \quad \forall (i,j) \in E, z \tag{2.2}$$

$$h_{ij} - h_{ji} = 0 \quad \forall (i,j) \in E$$

$$x_{ij}^{(z)} \geq 0 \quad \forall (i,j) \in E, z$$

$$h_{ij} = \{0,1\} \quad \forall (i,j) \in E \tag{2.3}$$

Since we consider a connectivity problem, only unit flow need to be carried from s to d, which is guaranteed by the flow constraints (2.1). Constraint (2.2) guarantees that in case failure z occurs and affects the link ($t_{ij}^{(z)} = 1$), unit flow can be carried on link (i,j) only if it is shielded ($h_{ij} = 1$). If link (i,j) is not affected by failure z ($t_{ij}^{(z)} = 0$), it can carry unit flow regardless of shielding in case of failure z. The factor of $1/2$ in the objective accounts for the fact that each shielded link is counted twice ($h_{ij} = h_{ji} = 1$).

The above algorithm can be applied to obtain the optimal shielding under the general failure model. For example, in Fig. 2-1, which represents the topology of the XO communication backbone network and consists of 60 nodes and 71 links [50], we consider the failure model where all the links incident to any two nodes are affected by a failure, and the number of failures is $\binom{60}{2}$ (i.e., this model allows for link failures incident to up to 2 nodes). The cost of shielding each link is represented by the length of the link (in latitude/longitude degree unit). Given the SD pair Seattle-Miami, the optimal shielding obtained by the algorithm is represented by the thick links with

total cost 45.98.

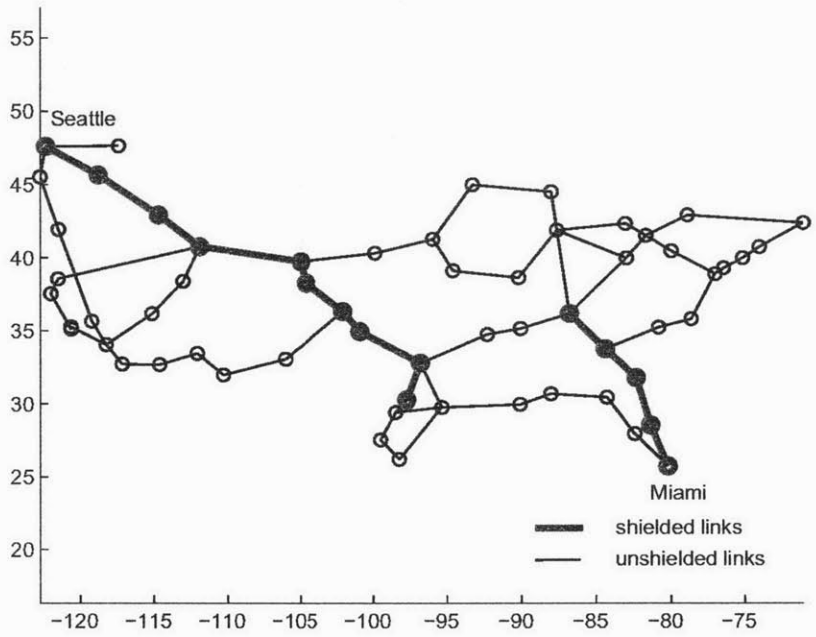

Figure 2-1: Optimal shielding under the failure model where all the links incident to any two nodes are affected by a failure.

2.2.2 Shielding under the geographical failure model

We model geographical failures as disks with a given radius (i.e., all links intersected by the disk region fail). Given a network topology and an SD pair, it is necessary to identify the set of geographical failure regions that disconnect the given SD pair. We call such regions bottleneck regions. Finding the bottleneck regions can be accomplished by checking whether the disk failure disconnects the SD pair [35]. Since the number of dominating failure regions is polynomial in the number of links [36, 1], this task can be done in polynomial time. In the following we exploit properties of bottlenecks to decompose the shielding problem to several subproblems, each of which consists one or more bottlenecks and can be solved separately.

In order to guarantee a path between the given SD pair, within each bottleneck a

shielded path that starts and ends outside the failure region must exist. Otherwise, there would be no path going through this bottleneck, and removing all the unshielded links within the bottleneck would leave the SD pair disconnected. If the shielded path is part of a path between the SD pair, such a shielded path is sufficient to guarantee that the SD pair is connected after a disk failure occurs at this bottleneck.

We start by describing a simple algorithm that can be used to find a shielded path to guarantee the connectivity of an SD pair after a failure occurs at a bottleneck. We illustrate the algorithm by using the example in Fig. 2-2.

Algorithm 1 Bottleneck Shielding Algorithm

1. Within the bottleneck, find the links that are connected to the source node without going through any link in the bottleneck (links a, b in Fig. 2-2). A link is connected to the source if a path exists between the source and one end node of the link. These links have to cross the boundary of the failure region. Among the two end nodes of each link, there is one node outside the failure region. Merge these nodes to form a dummy source, as shown in Fig. 2-2.
2. Find the links that are connected to the destination without going through any link in the bottleneck (links c, d in Fig. 2-2), and merge their ends outside the failure region to form a dummy destination.
3. Shield a path between the dummy SD pair. This path will survive the failure affecting this bottleneck.

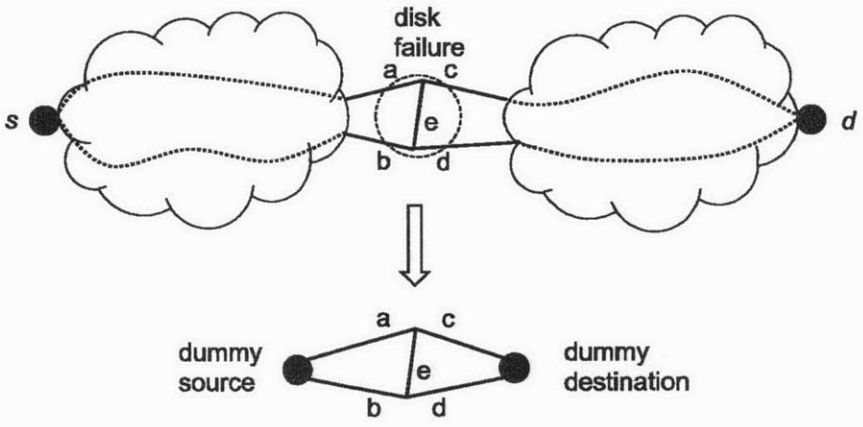

Figure 2-2: A single bottleneck.

Based on previous analysis, we have the following claim.

Claim 1. *A shielded path between the dummy SD pair is necessary and sufficient to guarantee the original SD pair connectivity after a disk failure at this bottleneck.*

If there is only one bottleneck between an SD pair, we only need to shield a "shortest path" between the dummy SD pair, where the "length" of each link represents its shielding cost. However, generally there may be multiple bottlenecks. Recall that each bottleneck includes a set of links which can fail simultaneously and whose failure disconnects the SD pair. In order to guarantee the connectivity of the SD pair in case of any disk failure, it is necessary to shield a path through every bottleneck. If the bottlenecks are disjoint and do not share common links (Fig. 2-3), shielding the shortest path between each dummy SD pair is optimal.

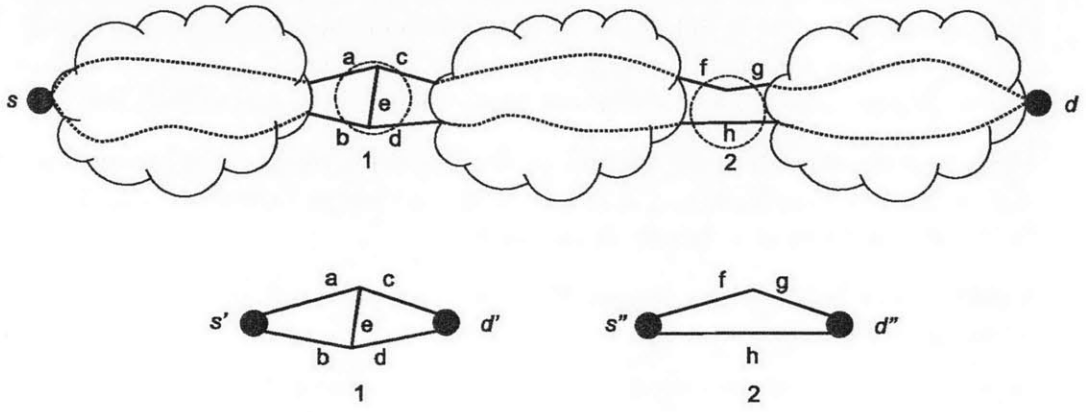

Figure 2-3: Non-overlapping bottlenecks.

However, different bottlenecks may overlap and share common links (Fig. 2-4). Shielding the links in one bottleneck may affect the shielding for another bottleneck. For example, between the first dummy SD pair (s', d'), shielding link c also leads to a shielded link c between the second dummy SD pair (s'', d''). Thus it is necessary to consider all the overlapping bottlenecks jointly.

Nevertheless, if a set of overlapping bottlenecks do not share common links with another set of overlapping bottlenecks, these two sets of bottlenecks can be considered separately, because shielding decisions for one set do not affect the shielding decisions for the other in order to shield a path in each bottleneck. The MILP (2.3) can be modified to determine the optimal shielding for a set of overlapping bottlenecks

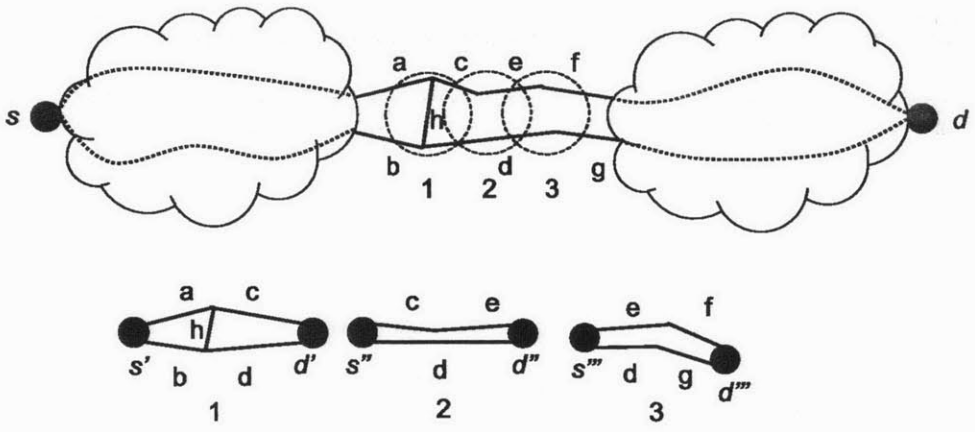

Figure 2-4: Overlapping bottlenecks.

which do not share common links with the other bottlenecks. Let $OB^{(w)}$ be a set of overlapping bottlenecks, and $E^{(w)}$ be the set of links intersected by $OB^{(w)}$, where w is an index for the overlapping bottlenecks set. The following MILP considers the possible failures $z \in OB^{(w)}$ to determine which links from $E^{(w)}$ should be shielded.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in E} c_{ij} h_{ij}/2 \\
\text{s.t.} \quad & \sum_{\{j|(i,j)\in E\}} x_{ij}^{(z)} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(z)} = \begin{cases} 1, \text{ if } i = s \\ -1, \text{ if } i = d \\ 0, \text{ otherwise} \end{cases} \quad \forall z \in OB^{(w)} \\
& x_{ij}^{(z)} - h_{ij} \leq 1 - t_{ij}^{(z)} \quad \forall (i,j) \in E^{(w)}, z \in OB^{(w)} \\
& h_{ij} - h_{ji} = 0 \quad \forall (i,j) \in E^{(w)} \\
& x_{ij}^{(z)} \geq 0 \quad \forall (i,j) \in E, z \in OB^{(w)} \\
& h_{ij} = \{0,1\} \quad \forall (i,j) \in E^{(w)}
\end{aligned}
\tag{2.4}
$$

The optimal shielding problem can be decomposed to subproblems which shield each disjoint bottleneck by a shortest path, and shield each set of overlapping bottlenecks using MILP (2.4). It is worth noting that the constraints in MILP (2.4) are the same as the constraints in MILP (2.3) associated with overlapping bottlenecks

$OB^{(w)}$. Under the geographical failure model, instead of considering all the failures as in MILP (2.3), the problem can be decomposed to multiple smaller MILPs (one per overlapping bottlenecks set) that can be solved more efficiently.

We illustrate the algorithm using the network in Fig. 2-5, which is the same as in Fig. 2-1, where now a failure is any disk with radius 2° (about 120 miles). Given the SD pair Seattle-Miami, there are 4 bottleneck regions represented by dashed circles, and all the links intersected by the dashed circles are candidate links among which shielding decisions are made. In each of the two disjoint bottlenecks, a shortest path is shielded, represented by the thick links with costs 5.12 and 5.88, respectively, while the overlapping bottleneck has shielding cost 10.00, yielding to a total shielding cost 21.00.

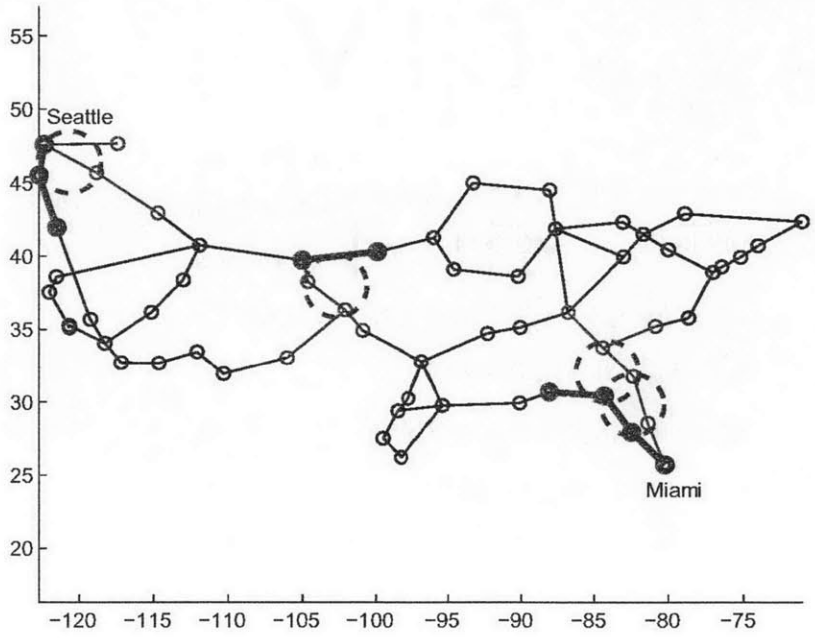

Figure 2-5: Bottlenecks and shielded links given disk failure radius 2° in the XO network.

2.3 Connectivity of the entire network

In backbone networks, where nodes represent routers, it is important to guarantee that all the nodes are connected. Since links are shared by many SD pairs, shielding links to guarantee the connectivity of one SD pair may benefit another SD pair. The union of the optimal shielding for each SD pair may not be the optimal shielding for the network. In fact, all the SD pairs must be considered jointly in order to determine the optimal shielding.

2.3.1 Shielding under the geographical failure model - the cases of huge and tiny failures

We start by considering two special cases of geographical failures. In the first case, the failure region is huge and contains all the links in the network. Unshielded links are destroyed by a failure event. In this case, in order to keep all the nodes connected, one must shield a spanning tree in the graph. The minimum cost shielding is a minimum weight spanning tree, where the weight of each link is its shielding cost.

In the second case, the failure region is tiny, which affects either a single link, or the links incident to one node. If the network has tree structure, every link's failure would disconnect the network. Therefore all the links in the tree structure have to be shielded. On the other hand, if nodes form a cycle, removing the links incident to one node does not affect the connectivity of the other nodes (recall that we only consider one failure at a time). Therefore, it is sufficient to guarantee that each node is incident to at least one shielded link, which connects this node with the remaining nodes. For a cycle structure, the optimal set of links to shield is the minimum edge cover. Minimum edge cover of a graph is a set of edges of minimum weight such that every node in the graph is incident to at least one edge in the set. The calculation of minimum edge cover takes polynomial time and can be obtained by calculating the maximum matching in a transformed graph [41].

It is possible that a network contains multiple cycles. Not all cycles can be considered independently. Instead, the minimum edge cover is calculated in each cycle com-

ponent, which is the union of cycles that are connected by common links. For example, the left graph in Fig. 2-6 is a single cycle component, since cycles $\{AC, AD, CD\}$ and $\{AB, AD, BD\}$ share a common link AD, and cycles $\{AB, AD, BD\}$ and $\{BD, BE, DE\}$ share a common link BD. If two cycles are connected by a common node but do not share links, they are separate cycle components. The right graph in Fig. 2-6 has two cycle components $\{AC, AD, CD\}$ and $\{BD, BE, DE\}$. For each cycle component, shielding the links in the minimum edge cover is optimal. To see this, if all the links incident to one node are removed, the remaining nodes in a cycle component are still connected. Therefore, a shielded link incident to every node is sufficient to guarantee its connectivity after any failure. It is obviously necessary since if no shielded link is incident to a node, the node would be disconnected from the other nodes in the cycle component after the failures of its incident links in the cycle component.

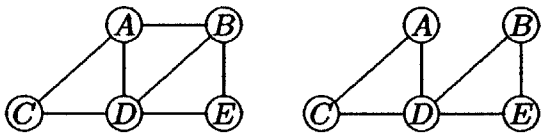

Figure 2-6: Illustration of cycle components. The left graph has one cycle component, while the right graph has two.

Different cycle components need to be considered separately, and the optimal shielding is the union of optimal shielding for each cycle component. This holds because shielding links in one cycle component does not benefit the connectivity of another cycle component. Therefore, in each cycle component, it is necessary that every node is incident to at least one shielded link. Moreover, such shielding is sufficient to guarantee that the graph is connected, since different cycle components are connected by nodes.

Based on the above development, we have the following algorithm which can be used to determine the optimal shielding for a given network under the tiny disk failure model.

Algorithm 2 Optimal Shielding Under the Tiny Disk Failure Model

1. Determine tree structures and cycle components in a graph (see Appendix for an algorithm to identify cycle components).
2. Shield all the links in tree structures.
3. For each cycle component, compute its minimum edge cover, where the weight of each edge is its shielding cost. Shield the links belonging to the minimum edge cover.

2.3.2 Shielding under the general failure model

Finally we consider optimal shielding to guarantee that the entire network is connected under the general failure model. The network is represented by a graph $G = (V, E)$. Each failure z affects a set of links $E^{(z)}$. Our objective is to shield a set of links E^* that have minimum shielding cost, to guarantee that $G = (V, E'^{(z)})$ is connected for all z, where $E'^{(z)} = (E \setminus E^{(z)}) \cup E^*$.

The MILP formulation for this problem is similar to the formulation for the single SD pair connectivity problem, except that the constraints guarantee the connectivity of the entire network instead of a single SD pair. The variables and parameters have the same meaning as in MILP (2.3).

$$\min \quad \sum_{(i,j)\in E} c_{ij} h_{ij}/2$$

$$\text{s.t.} \quad \sum_{\{j|(i,j)\in E\}} x_{ij}^{(z)sd} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(z)sd} = \begin{cases} 1, \text{ if } i = s \\ -1, \text{ if } i = d \\ 0, \text{ otherwise} \end{cases} \quad \forall z, s, d$$

$$x_{ij}^{(z)sd} - h_{ij} \leq 1 - t_{ij}^{(z)} \quad \forall (i,j) \in E, z, s, d \tag{2.5}$$

$$h_{ij} - h_{ji} = 0 \quad \forall (i,j) \in E$$

$$x_{ij}^{(z)sd} \geq 0 \quad \forall (i,j) \in E, z, s, d$$

$$h_{ij} = \{0, 1\} \quad \forall (i,j) \in E \tag{2.6}$$

In MILP (2.6), for each SD pair and failure scenario, there is a flow variable for each link. The number of variables is very large since there are many possible failure scenarios. It is difficult to directly solve MILP (2.6) for large problem instances. However, the flow variables after one failure couple with the flow variables after another failure only through the decision variables h in (2.5). Given h, it is easy to determine whether there are feasible flows between all the SD pairs after each failure, by only considering the flow variables and constraints related to each failure. Benders decomposition can be applied to problems with such partial separable structure.

Benders decomposition

Benders decomposition accelerates the calculation of an optimization problem with partial separable structure and many constraints [5]. Instead of considering all the constraints at the same time, it first solves a relaxed problem that has only a few constraints, and then check whether there are any violated constraints. If there are no violated constraints, the solution is optimal. Otherwise, if a violated constraint is identified, it is added to the relaxed problem and the problem is solved again. The relaxed problem is called the master problem, and violated constraints are identified by solving subproblems.

The MILP (2.6) can be reformulated as follows. It starts with a master problem with constraints only on h.

$$\begin{aligned} \min \quad & \sum_{(i,j)\in E} c_{ij}h_{ij}/2 \\ \text{s.t.} \quad & h_{ij} - h_{ji} = 0 \ \ \forall (i,j) \in E \\ & h_{ij} = \{0,1\} \ \ \forall (i,j) \in E \end{aligned}$$

After obtaining h, check whether there are violated constraints by solving subproblems, each corresponding to checking whether the network is connected after each failure. If the linear program (LP) (2.7) is feasible and has optimal value 0, the

network is connected after failure z. If it is infeasible, the associated constraint has been violated.

$$
\begin{aligned}
\min \quad & 0 \\
\text{s.t.} \quad & \sum_{\{j|(i,j)\in E\}} x_{ij}^{(z)sd} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(z)sd} = \begin{cases} 1, \text{ if } i = s \\ -1, \text{ if } i = d \\ 0, \text{ otherwise} \end{cases} \quad \forall s, d \\
& x_{ij}^{(z)sd} - h_{ij} \leq 1 - t_{ij}^{(z)} \quad \forall (i,j) \in E, s, d \\
& x_{ij}^{(z)sd} \geq 0 \quad \forall (i,j) \in E, s, d
\end{aligned}
\tag{2.7}
$$

It is more efficient to add the violated constraint by considering the dual of LP (2.7). The dual is represented by LP (2.8). If LP (2.7) is infeasible, LP (2.8) is unbounded, and the constraint $\sum_{sd}[p_d^{*(z)sd} - p_s^{*(z)sd} - \sum_{(i,j)\in E}(1 - t_{ij}^{(z)} + h_{ij})r_{ij}^{*(z)sd}] \leq 0$ is added to the master problem, where $(p_d^{*(z)sd}, p_s^{*(z)sd}, r_{ij}^{*(z)sd})$ is an extreme ray of LP (2.8). The constraints avoid unbounded costs along extreme rays, to guarantee that LP (2.8) is bounded and that LP (2.7) is feasible.

$$
\begin{aligned}
\max \quad & \sum_{sd}[p_d^{(z)sd} - p_s^{(z)sd} - \sum_{(i,j)\in E}(1 - t_{ij}^{(z)} + h_{ij})r_{ij}^{(z)sd}] \\
\text{s.t.} \quad & p_j^{(z)sd} - p_i^{(z)sd} - r_{ij}^{(z)sd} \leq 0 \quad \forall (i,j) \in E, s, d \\
& r_{ij}^{(z)sd} \geq 0 \quad \forall (i,j) \in E, s, d
\end{aligned}
\tag{2.8}
$$

While for the original problem, the MILP has a large number of variables and constraints, with Benders decomposition it is possible to solve each subproblem using an LP, and the size of each subproblem is small. Moreover, a slight modification of the standard Benders decomposition can better accommodate our problem instances. In the standard Benders decomposition, the violated constraints corresponding to one subproblem are added before solving the master problem in an iteration. Instead,

the violated constraints corresponding to multiple subproblems can be added in each iteration for the following reason. In our problem, checking whether a subproblem is bounded corresponds to checking whether the network is connected after one failure. Instead of checking only one failure as in the standard Benders decomposition, multiple failures can be checked in each iteration. This is particularly helpful since different failures affect different links, and the links that need to be shielded are likely to be different. The number of violated constraints added before resolving the master problem provides a tradeoff between the number of master iterations and the running time of each iteration. In our numerical evaluations, the number of constraints that we added was equal to the number of nodes in the network, and we observed more than 50% running time saving compared with the standard Benders decomposition algorithm.

Simulated annealing

Finally, we developed a heuristic based on simulated annealing [4, 25] to solve the problems faster. Simulated annealing is a method to search for globally optimal solutions for nonconvex optimization problems. It starts at an initial state, and then aims to find a neighbor state, preferably a state with smaller cost. If such a neighbor state with smaller cost is found, the current state is replaced with the neighbor state. Otherwise, if the neighbor state has larger cost, the current state is replaced with the neighbor state with some small probability. Simulated annealing avoids being stuck in a local minimum without continuing further searches. The probability to replace the current state with a neighbor state that has higher cost depends on the difference of the costs of the two states. The higher the cost of the neighbor state, the less likely to enter this state.

Let S be the set of shielded links and S^c be the set of unshielded links in the current state. Initially, all the links are shielded. In order to find a neighbor state which differs from the current state in only one or two shielded links, one of the following three operations is performed:

1. Randomly remove one link from S.

2. Randomly remove one link from S, and randomly shield one more link from S^c.

3. Randomly shield one more link from S^c.

In these operations, the probability of removing a link is proportional to the shielding cost of the link. Thus, links having larger shielding costs are more likely to be removed. The probability of adding a link to the shielded set is proportional to the multiplicative inverse of its shielding cost, so that links with small shielding costs are more likely to be added.

Since the objective is to find a neighbor state with smaller shielding cost, the operations are done sequentially during the first few iterations of simulated annealing. For example, after one shielded link is removed (operation 1), if the current shielding is feasible, a neighbor state with smaller shielding cost is identified. If any removal of shielded link leads to an infeasible shielding, one more shielded link is added after the removal (operation 2) in search of a feasible shielding state. If neither works, one more shielded link is added without removing any shielded link (operation 3), in order to retain a feasible shielding state.

After finding the neighbor state, next determine whether to replace the current state with the neighbor state. If the neighbor state has smaller shielding cost than the current state, it replaces the current state. On the other hand, if the neighbor state has larger shielding cost compared with the current state, it replaces the current state with probability $exp(-\delta/T)$, where δ is the difference between the shielding costs of the two states and T is the temperature. It is possible to enter a state with higher shielding cost and avoid being stuck at a local minimum.

During the first fewer iterations, T is large so that it is easy to enter a state that has larger shielding cost to explore more possible states. As the number of iterations increases, T decreases to make it less likely to enter a state with larger shielding cost. At last, T is decreased to 0 and the algorithm terminates at a state which has smaller shielding cost compared to all its neighbors. The optimal decrease of T in theory follows $T(t) = d/\log t$ and is adopted in our algorithm, where d is a positive constant and t is the number of iterations [25].

If the operations to find a neighbor state are done sequentially, the algorithm may end up in cycles. It is possible that both operations 1 and 2 cannot find a feasible shielding for the network. The only possible next state is to shield an extra link. Starting from the next state, the only link that can be removed without causing infeasible shielding is the link that was just added. Therefore, after some iterations when many redundantly shielded links are removed, the operations need to be done randomly to avoid such cycles.

2.4 Guaranteeing partial connectivity

In most networks, a significant number of links need to be shielded to guarantee the full connectivity of the network. In fact, even in the tiny disk failure case, links need to be shielded according to the minimum edge cover for cycle structures. The number of shielded link is at least half the number of nodes. In larger failure cases or if the network has tree structures, even more links need to be shielded.

If the connectivity constraint is relaxed and some nodes are allowed to be separated from the others, shielding cost may be significantly reduced. The reduction in shielding depends on the failure model and network topology. For example, if one node is allowed to be disconnected from the rest, in cycle structures no link need to be shielded in the tiny disk failure case, since only the node within the failure region is disconnected from the others. Similarly, in tree structure, links incident to degree 1 nodes do not need to be shielded, since the failure of a link incident to degree 1 node only separates a degree 1 node from the others.

We determine the optimal shielding to guarantee partial network connectivity under the general failure model, using average two terminal reliability (ATTR) as a measure of the connectivity level. ATTR is calculated by dividing the number of connected SD pairs after a failure by the total number of SD pairs in the original network, and represents the fraction of SD pairs connected after a failure. Compared to MILP (2.6), the unit flow constraints are not imposed to every SD pair. Instead, only a fraction of SD pair are guaranteed to carry unit flow. In constraints (2.9), I^{sd} can either take the value 0 or 1, where $I^{sd} = 1$ guarantees the connectivity of the SD

pair. The total number of connected SD pair should be at least a fraction α of all the $N(N-1)/2$ SD pairs, guaranteed by constraints (2.10), where N is the total number of nodes.

$$\min \sum_{(i,j)\in E} c_{ij}h_{ij}/2$$

$$\text{s.t.} \sum_{\{j|(i,j)\in E\}} x_{ij}^{(z)sd} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(z)sd} = \begin{cases} I^{(z)sd}, \text{ if } i = s \\ -I^{(z)sd}, \text{ if } i = d \\ 0, \text{ otherwise} \end{cases} \quad \forall z, s, d \tag{2.9}$$

$$x_{ij}^{(z)sd} - h_{ij} \leq 1 - t_{ij}^{(z)} \quad \forall (i,j) \in E, z, s, d$$

$$\sum_{sd} I^{(z)sd} \geq \alpha N(N-1)/2 \quad \forall z \tag{2.10}$$

$$h_{ij} - h_{ji} = 0 \quad \forall (i,j) \in E$$

$$x_{ij}^{(z)sd} \geq 0 \quad \forall (i,j) \in E, z, s, d$$

$$h_{ij} = \{0,1\} \quad \forall (i,j) \in E$$

$$I^{sd} = \{0,1\} \quad \forall (i,j) \in E$$

(2.11)

The above MILP has more variables and constraints than MILP (2.6). First, there are the additional variables I^{sd}. Moreover, in MILP (2.6), we only need to check the connectivity between node 1 and nodes $2, 3, \ldots, N$, which is enough to guarantee the connectivity of the entire network. However, in MILP (2.11), we need to check $N(N-1)/2$ SD pairs. We again use simulated annealing to find near-optimal solutions. The only difference compared with the algorithm which guarantees full connectivity is in determining whether the shielding is feasible. As long as the ATTR is above α, the shielding is feasible and is a candidate for the next state.

2.5 Numerical results

2.5.1 Full connectivity

We first compare the running time of solving the MILP using Benders decomposition, solving the MILP directly, and solving its LP relaxation in Fig. 2-7. The results are averaged over 10 instances of Erdos-Renyi random graphs. The number of nodes of the graph is varied from 10 to 30, with average degree 5. We consider failures that affect the links adjacent to two nodes. The number of possible attacks for each graph is $\binom{N}{2}$, where N is the number of nodes in a graph. Note that solving the MILP directly and solving the MILP using Benders decomposition both give the optimal solutions, while solving the LP relaxation only gives lower bounds of optimal shielding costs. It can be observed that solving the MILP using Benders decomposition works even faster than solving the LP relaxation of the MILP directly for larger networks.

Moreover, the modified Benders decomposition reduces the running time further by more than 50% in most cases as shown in Table 2.1.

Next we compare the performance of simulated annealing with the modified Benders decomposition. We observe in Tables 2.1 and 2.2 that the running time for simulated annealing is about 1/10 of that of modified Benders decomposition in larger network cases, while the relative error is only $3\% \sim 6\%$.

Table 2.1: Running time comparisons of SA, BD and the modified BD for random graphs

Number of nodes	Degree	SA Time (s)	BD Time (s)	Modified BD Time (s)
10	5	0.79	2.26	1.54
15	5	1.70	20.89	10.85
20	5	3.96	90.77	34.96
25	5	10.86	270.69	103.32
30	5	19.20	684.83	195.20

Finally we apply our algorithm to obtain the optimal shielding for the XO communication network. Fig. 2-8 illustrates the optimal shielding to guarantee the connectivity of the entire XO backbone network after any disk failure with radius 1°.

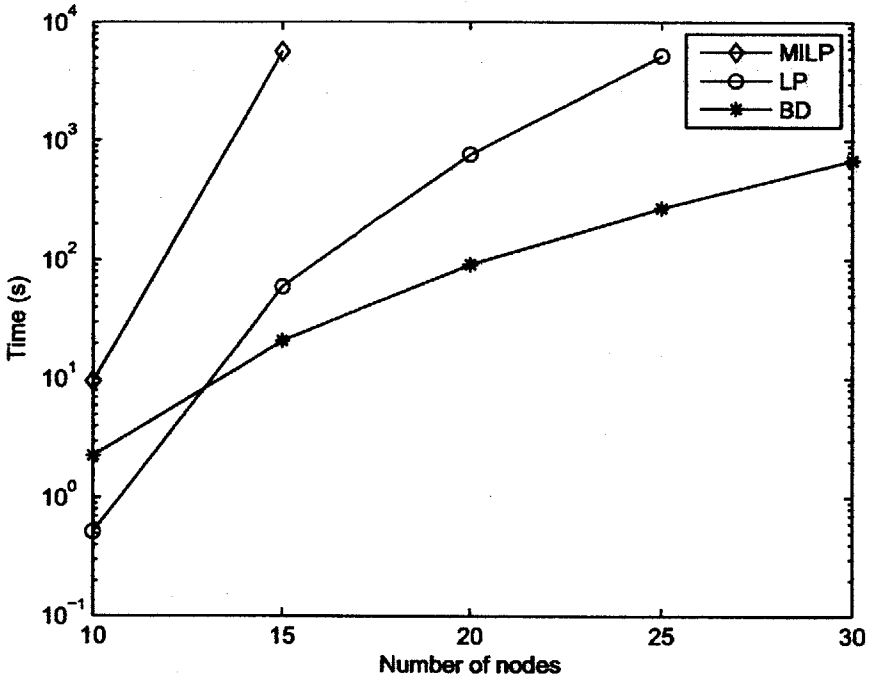

Figure 2-7: Running time comparisons of Benders decomposition, directly solving MILP and its LP relaxation.

Simulated annealing also has good performance in solving the shielding problem for the XO network. The results are shown in Table 2.3.

2.5.2 Partial connectivity

The MILP for partial connectivity involves a large number of variables and constraints, and can only be solved for small problem instances (were able to solve for networks which have fewer than 15 nodes). On the other hand, simulated annealing algorithm which guarantees partial connectivity has comparable running time with the algorithm which guarantees full connectivity. The only difference between the two simulated annealing algorithms is in determining whether the shielding is feasible. The shielding costs obtained by the simulated annealing algorithm are nearly identical with the exact solutions by solving the MILP for different levels of ATTR

Table 2.2: Shielding cost comparisons of SA results and exact solutions for random graphs

Number of nodes	Degree	SA Cost	Exact Cost	Relative Error
10	5	67.2	64.8	0.037
15	5	141.4	136.2	0.038
20	5	248.0	240.0	0.033
25	5	394.4	374.2	0.054
30	5	551.4	532.8	0.035

Table 2.3: Comparison of SA and the Modified BD algorithms for the XO network

Attack radius	SA Time (s)	Modified BD Time (s)	SA Cost	Modified BD Cost	Relative error
1°	51.98	109.19	106.6	99.5	0.071
2°	64.73	875.12	129.3	121.3	0.065

requirement, and are omitted.

Shielding cost is significantly reduced by relaxing the connectivity constraint to allow $\alpha = 95\%$. This corresponds to the case that one node can be disconnected from the others in the XO network ($\alpha = 29/30 > 95\%$). Figure 2-9 depicts the shielded links in the case where the disk failure has radius 1°. Table 2.4 suggests that the reduction depends on the failure model, with larger reduction for smaller failure.

Table 2.4: Cost reduction of partial connectivity for the XO network

Attack radius	SA Cost (full connectivity)	SA Cost ($\alpha = 95\%$)
1°	106.5	26
2°	129	86

2.6 Conclusion

In this chapter we considered the network shielding problem under geographical and general failure models. We developed MILP formulations to obtain the minimum cost shielding to guarantee the connectivity of a single SD pair and the entire network under the general failure model. To guarantee the connectivity of a single SD pair under the geographical failure model, we develop an algorithm to decompose the problem

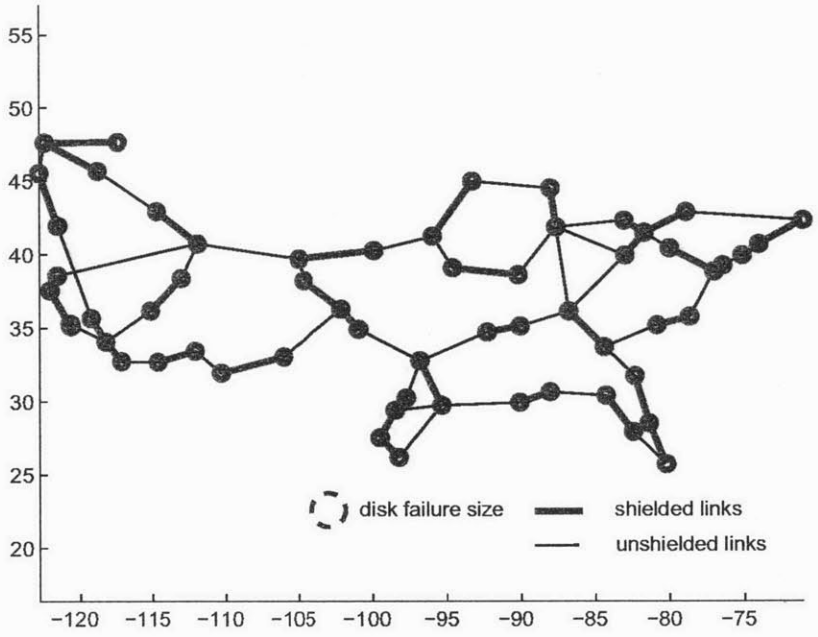

Figure 2-8: Optimal shielding in the XO network to guarantee full connectivity after any disk failure with radius 1°.

to multiple subproblems, each of which determines the optimal shielding for links in a geographical region. The MILP that guarantees the connectivity of the entire network has partial separable structure, for which Benders decomposition can be applied to significantly reduce the running time. A slightly modified Benders decomposition reduces the running time further by more than 50%. In addition, simulated annealing is used to obtain near-optimal solutions with much shorter running time.

Significantly less shielding cost is required to guarantee partial connectivity, even in the case where only one node is allowed to be disconnected from the others. Moreover, we observe larger reduction in shielding cost if the size of a failure region is small. The algorithms can be easily modified to solve the problem which guarantees the connectivity of a selected set of SD pairs in a network. For example, in the MILP, the flow constraints can be imposed only for the selected set of SD pairs. The methodologies in this chapter can be used to construct new and upgrade existing networks to guarantee connectivity after any single geographical or general failure.

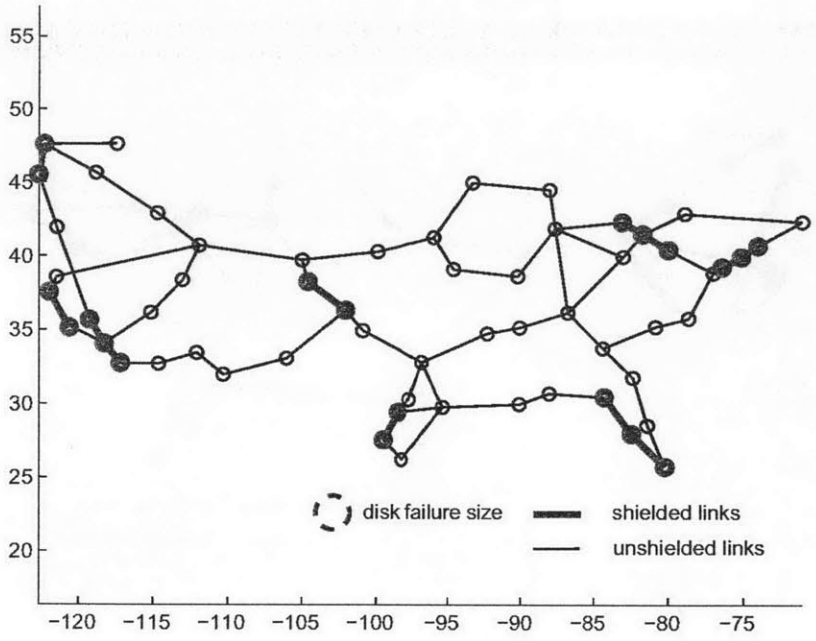

Figure 2-9: Optimal shielding in the XO network to guarantee that $\alpha = 95\%$ after any disk failure with radius 1°.

2.7 Chapter Appendix: an algorithm to find cycle components

The following is an algorithm that finds cycle components of a graph. First, determine the links in tree structures. A link belongs to the tree structure if the network is disconnected after the removal of the link.

After all the links belonging to tree structures are removed, one or more connected graphs remain, denoted by $G_1, G_2, \ldots, G_k$. Different connected graphs do not share the same nodes or links, and can be considered separately.

For each G_i, cycles that share the same links need to be identified to obtain cycle components. Instead of considering all the cycles, which can be exponential in the number of links, only the cycles in a cycle basis needs to be considered. A cycle basis is a minimal set of cycles that can be used to form all the cycles in a graph [46]. It

is easy to see that if two cycles share a common link, the corresponding cycles in the basis also share at least one common link. Therefore, all the cycles that share common links can be identified from the cycle basis. Cycle components can be obtained by merging the cycles in a cycle basis that share common links.

To find a cycle basis, a spanning tree is first identified, and each of the remaining links is a back link. Each back link together with the links in the spanning tree defines a cycle in a cycle basis [46]. If there are $|V_i|$ nodes and $|E_i|$ links in G_i, the number of links in a spanning tree is $|E_i|-1$, and the cardinality of a cycle basis is $|V_i|-|E_i|+1$.

We illustrate the algorithm in Fig. 2-10. Links CD and FG belong to tree structures. After removing them, there are two connected graphs. One is formed by nodes $\{D, G, H\}$ and links connecting them, and the other is $\{A, B, E, F\}$. Obviously, there is only one cycle (component) containing $\{D, G, H\}$. In the graph containing $\{A, B, E, F\}$, there exists a spanning tree $\{AB, AE, AF\}$. Each remaining link defines a cycle in a basis, i.e., $\{AE, AF, EF\}$, $\{AB, AF, BF\}$, and $\{AB, AE, BE\}$. Since $\{AE, AF, EF\}$ and $\{AB, AF, BF\}$ share a common link AF ($\{AE, AF, EF\}$ and $\{AB, AE, BE\}$ share a common link AE), the nodes $\{A, B, E, F\}$ and the links connecting them form a cycle component.

Depth first search can be used both to check whether a graph is connected and to obtain the cycle basis. The complexity to do a depth first search in $G = (V, E)$ is $O(|E|)$. Finding all the links belonging to tree structures takes $O(|E|^2)$ by checking whether the graph is connected after removing each of the $|E|$ links. After removing all the links in tree structures, the larger a remaining connected graph, the more difficult to find the cycle components. In the worst case, no link is removed and the cardinality of a cycle basis is at most $|E| - |V| + 1$. Checking whether two or more cycles share a common link and identifying cycle components in the remaining graph take $O(|E|(|E|-|V|+1)) = O(|E|^2)$ time. Therefore, the complexity of the algorithm to find cycle components is $O(|E|^2)$.

The optimal shielding is to shield all the links in tree structures and links in minimum edge cover for each cycle component. In Fig. 10, these links are AB, EF, FG, GH, DH, and CD.

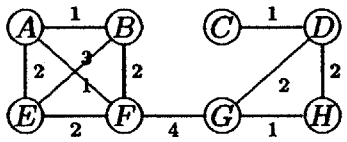

Figure 2-10: Two links in tree structures and two cycle components.

Chapter 3

Shielding to increase network effective min-cut

A cut in a graph is a set of links whose removal separates the nodes into two disjoint sets. A cut of minimum size is a minimum cut (min cut), whose size is denoted by the min-cut and is an important measure of network resilience. The larger the min-cut, the more link failures that the network can resist. Small cuts are "bottlenecks" of the network and should be avoided in the design of robust networks.

Link augmentation can be used to eliminate small cuts in a graph and to increase its min-cut. Given a graph with existing links, previous works develop polynomial time algorithms to augment new links with minimum cost such that an SD pair is k connected, which can be solved as a k link disjoint shortest paths problem by assigning zero weight to the existing links [45]. In order to guarantee that the graph is k connected, determining the minimum number of links that need to be augmented is also polynomially solvable [48], given that links can be constructed between any two nodes of the graph. It is also possible to augment links by selecting from a specified set, e.g., links not in the set can be assigned arbitrarily high costs such that they cannot be selected in the augmentation. However, the general augmentation problem, which determines the minimum cost augmentation to construct a k connected graph where each link is associated with an arbitrary cost, is NP-hard [22]. Mathematical programming techniques of developing stronger formulations by adding cutting

planes and finding facet defining inequalities, are usually used to solve the general augmentation problem [27, 44].

In this chapter we consider eliminating small cuts by shielding links, where shielded links are assumed to be perfectly reliable and cannot be included in a cut. We refer to cuts that do not contain shielded links as "effective cuts" and aim to increase the effective min-cut of the network by shielding. Shielding a link can also be viewed as augmenting a sufficiently large number of parallel links, such that these parallel links cannot be in any cut of size less than k in the augmented graph which has effective min-cut k.

In this chapter, we first describe the random failure model, where increasing the effective min-cut is key to improving network reliability. Then we consider shielding links to increase the effective min-cut for a single SD pair. We develop a polynomial algorithm for the optimal shielding to increase the effective min-cut by 1, and develop a MILP to obtain the optimal shielding to increase the effective min-cut by an arbitrary value. Finally we consider increasing the effective min-cut for the entire network. We provide an upper bound on the minimum cost required to increase the network effective min-cut by 1, and extend the MILP to obtain the optimal shielding to increase the network effective min-cut by an arbitrary value.

3.1 Random link failure model

Random link failures occur in communication networks since network components may break down or malfunction. We use "failure" and "removal" interchangeably to denote the break down or malfunction of a link. There have been extensive works to determine network reliability under random link failures [29, 30]. Given the link failure probability, one approach is to use the reliability polynomial to determine the probability that a given SD pair or a network is disconnected.

Next we briefly discuss the reliability polynomial and see the importance of increasing the effective min-cut on network reliability. Given a graph $G = (V, E)$, a source node s and a destination node d, each link has failure probability p. Let N be

the total number of links, and m be the min-cut for the given SD pair. If fewer than m links fail, the SD pair is always connected. If m or more links fail, it is possible that the SD pair becomes disconnected. Let N_k be the number of combinations of k link failures that disconnect the SD pair. The reliability polynomial contains the term $N_k p^k (1-p)^{N-k}$, which is the probability that there are exactly k link failures and the SD pair is disconnected, and is given by:

$$F(p) = \sum_{k=m}^{N} N_k p^k (1-p)^{N-k}.$$

The reliability polynomial can also be used to evaluate the reliability of the entire network, where N_k is the number of combinations of k link failures that disconnect the network.

Shielded links are regarded as perfectly reliable and thus any cut that contains a shielded link is not counted in N_k. If the link failure probability p is small, terms with smaller values of k dominate $F(p)$. Increasing the effective min-cut significantly reduces $F(p)$ if $p \ll 1$.

3.2 Increasing the effective min-cut of an SD pair

To eliminate cuts $C = \{c_i | i = 1, \ldots, n\}$ in a graph, at least one link in each cut c_i need to be shielded, such that these shielded links cannot be contained in the effective cuts. There are two difficulties that make this direct approach inefficient. First, the number of cuts may be large, e.g., even the number of min cuts can be exponential in the number of links. Second, since each of these cuts should have at least one shielded link, finding the minimum cost shielding is an instance of minimum hitting set problem, where each element is a link and each set is a cut to be eliminated. The minimum hitting set problem determines a set of elements with minimum cost such that at least one element is in each set. Instead, we exploit the properties of min cuts between an SD pair to develop an efficient algorithm to eliminate all of them. Later we develop a MILP to eliminate small cuts whose size is below a given threshold.

3.2.1 Eliminating min cuts between an SD pair

A shielded path between an SD pair eliminates all the cuts that separate the SD pair. It is natural to expect that shielding the links in a path which belong to the min cuts eliminates all of the min cuts. Next we formally prove this theorem.

Theorem 1. *Given any path between an SD pair, shielding the links in that path that belong to min cuts of the SD pair is necessary and sufficient to eliminate all the min cuts.*

Proof. Suppose the min-cut between an SD pair is m. Eliminating all the cuts of size m is equivalent to shielding the links such that the SD pair is connected after any m unshielded links' removals. Any m link removals that disconnect the SD pair must not include a link which does not belong to min cuts. Therefore, a link which does not belong to min cuts exists in the residue graph after any m link removals that disconnect the SD pair, and does not need to be shielded. Shielding only the links belonging to min cuts in a path guarantees a path between the SD pair after any m link removals (recall that shielded links cannot be removed). Therefore, the shielding is sufficient to eliminate all the min cuts.

Next we prove that in order to eliminate all the min cuts, it is necessary to shield all the links belonging to min cuts in at least one path. Shielding a link in order that it cannot be in any cut is equivalent to increasing its capacity to infinity such that it cannot be in any finite cut. According to max-flow min-cut theorem, in the original graph the max-flow between the SD pair is equal to the min-cut m and there are m link disjoint paths. After eliminating all the min cuts of size m by shielding a set of links, the max-flow between the SD pair in the shielded graph is at least $m + 1$. Since there are only m link disjoint paths, an augmenting path that carries unit flow has to share common links which belong to min cuts with the existing paths. The common links need to be shielded such that they can carry more than one unit flow. Therefore, there exists at least one path such that the capacity of the links belonging to min cuts are shielded. □

Next we discuss how to identify links belonging to min cuts. Enumerating all the

min cuts is inefficient because the number of min cuts may be large. Instead, such links can be identified by first finding the max flow between the SD pair. The links belonging to min cuts have to be in the max flow paths. For each link in the max flow paths, it belongs to min cuts if and only if its removal reduces the max flow by one.

Based on previous analysis, the following algorithm can be used to shield a set of links with minimum cost to eliminate all the min cuts between an SD pair.

Algorithm 3 Algorithm to Eliminate Min Cuts Between an SD Pair

1. Identify the links belonging to min cuts. Set the "length" of each link which belongs to min cuts to its shielding cost. Set the "length" of any other link to zero.
2. Find a shortest path between the SD pair, and shield the links that have positive length.

We use the above algorithm to obtain the optimal shielding to eliminate all the min cuts between the SD pair Seattle-Miami in the XO network illustrated in Fig. 3-1. The min-cut between Seattle-Miami is 2 and the red links (grey in printed version) are the links belonging to min cuts. The optimal shielding that eliminates all the min cuts and increases the effective min-cut by one is represented by the thick links with total cost 35.18.

3.2.2 Eliminating small cuts between an SD pair

In addition to eliminating all of the min cuts, it is desirable to eliminate cuts with size smaller than certain threshold (i.e., small cuts), such that the SD pair remains connected after a large number of link failures. For example, in Fig. 3-2, the min-cut between the SD pair is 1. Shielding link a eliminates the min cut. To eliminate cuts of size 2, which guarantees that the SD pair is connected after any 2 link failures, either link b or link c need to be shielded in addition to link a. The problem of eliminating small cuts is a generalization of eliminating the min cuts. As we will see next, shielding the links belonging to small cuts in a path between an SD pair is

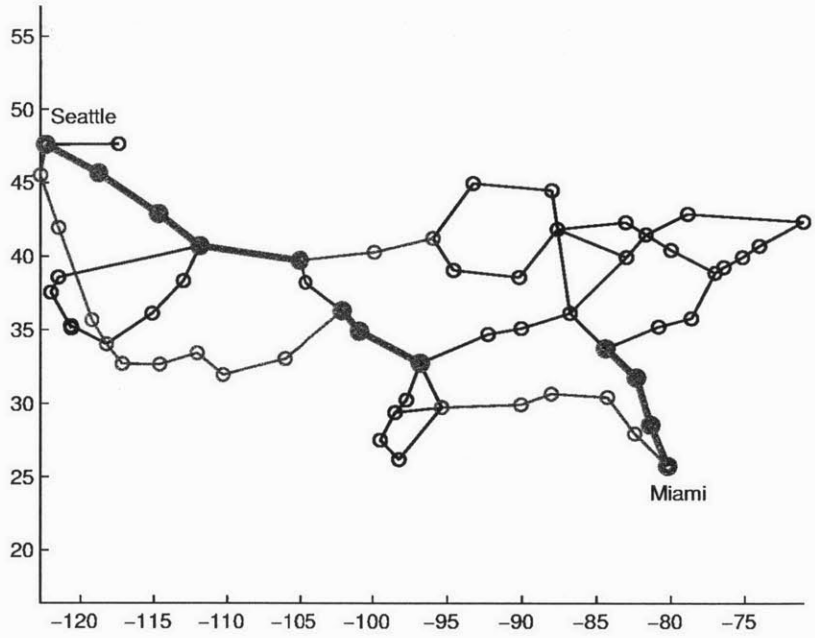

Figure 3-1: Optimal shielding to eliminate all the min cuts between an SD pair.

sufficient to guarantee that these cuts are eliminated, but it is no longer necessary.

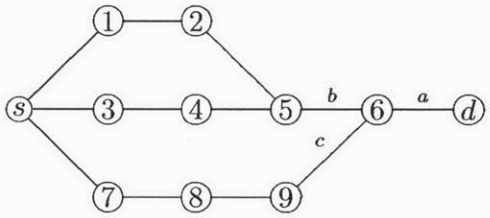

Figure 3-2: Shielding critical links to eliminate small cuts.

Theorem 2. *Consider the links in cuts of size k or less. Shielding such links in a path between an SD pair eliminates all the cuts of size k or less (i.e., the effective min-cut between the SD pair after shielding is at least $k+1$).*

Proof. The proof is similar to the sufficiency proof in Theorem 1. Eliminating all the cuts of size k or less is equivalent to shielding the links such that the SD pair is connected after any k unshielded links' failures. Let C denote the set of links which do not belong to cuts of size k or less. Links not in C do not need to be shielded,

because they can not be removed in case of k link failures that disconnect the SD pair. Shielding the links belonging to cuts of size k or less in a path guarantees a path between the SD pair after k link failures. Therefore, the shielding is sufficient to eliminate all the cuts of size k or smaller. □

In fact, only minimal cuts need to be considered among the cuts of size k or less. A minimal cut is a set of links where none of its proper subset is a cut. The k links whose failures disconnect the SD pair can be decomposed to a minimal cut M and additional links N (possibly $N = \emptyset$). Removals of links in any proper subset of M does not disconnect the network, while removals of all the links in M disconnects the SD pair regardless of the removals of links in N.

Corollary 1. *Consider the links in minimal cuts of size k or less. Shielding such links in a path between an SD pair eliminates all the cuts of size k or less.*

Proof. Shielding the links which belong to minimal cuts of size k or less in a path eliminates all these minimal cuts, because any minimal cut of size k or less would not disconnect the SD pair after the shielding. Moreover, any cut of size k or less is also eliminated, because all the minimal cuts contained in this cut are eliminated. Therefore, by shielding the links belonging to minimal cuts of size k or less, all the cuts of size k or less are eliminated. □

The following example in Fig. 3-3 shows that there may be significantly fewer links in minimal cuts. The only minimal cut that has size no larger than three is link a, while there are many combinations of three link removals that disconnect the SD pair. In fact, the set of any two links in a path between s and node 5 together with a, is a cut of size three that disconnects the SD pair. For example, links $\{a, b, c\}$ is an SD cut that separates nodes $\{d, 2\}$ from the others. Despite that all the links belong to cuts with size no larger than three, to resist any three link failures, only link a rather than a path between the SD pair needs to be shielded, since link a is the only link in the minimal cut with size no larger than three.

Nevertheless, to guarantee that the SD pair remains connected after k links failures, it is not necessary to shield the links belonging to minimal cuts of size k or less

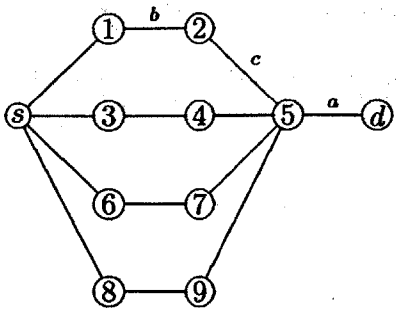

Figure 3-3: Illustration of minimal cuts (only link a is in minimal cuts of size $|C| \leq 3$).

in a path. To see this, consider the example in Fig. 3.2.2, where all the links belong to minimal cuts of size $|C| \leq 3$. To eliminates all the cuts of size $|C| \leq 3$, only links a, b and c need to be shielded, since any cut of size $|C| \leq 3$ contains one of the three links.

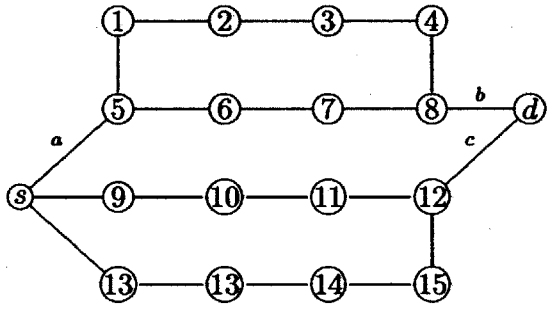

Figure 3-4: A shielded path is unnecessary (only links $\{a, b, c\}$ rather than a path need to be shielded to eliminates cuts of size $|C| \leq 3$).

Next we consider how to exactly obtain the minimum cost shielding to eliminate all the cuts of size smaller than F. To guarantee that shielded links cannot be in any effective cut, the capacity of the shielded link can be increased to sufficiently large such that any cut containing the link has capacity F or larger.

First consider a cut formulation that guarantees that the effective min-cut between an SD pair is at least F after shielding. Let $\{(i,j)|h_{ij} = 1\}$ be the set of shielded links. Let x_{ij} be the capacity of link (i,j), which is at most 1 for an unshielded link and may take a sufficiently large value up to $M + 1 \geq F$ for a shielded link.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in E} c_{ij}h_{ij}/2 \\
\text{s.t.} \quad & \sum_{(i,j)\in\delta(S)} x_{ij} \geq F \quad \forall S \subset V, S \neq V, s \in S, d \notin S && (3.1)\\
& x_{ij} \leq 1 + Mh_{ij} \quad \forall (i,j) \in E && (3.2)\\
& h_{ij} - h_{ji} = 0 \quad \forall (i,j) \in E \\
& x_{ij} \geq 0 \quad \forall (i,j) \in E, n \\
& h_{ij} = \{0,1\} \quad \forall (i,j) \in E \\
& && (3.3)
\end{aligned}
$$

Constraints (3.1) guarantee that any cut that separates s and d is at least F, where $\delta(S)$ denote the set of links that connect S and $V \setminus S$. The capacities of the links are limited by constraints (3.2), which guarantee that if link (i,j) is shielded, its capacity is increased to at most $M+1$. Note that the inequality constraints (3.2) can be replaced by equality constraints, since x_{ij} may take the largest possible value restricted by constraints (3.2) and satisfy constraints (3.1) given the feasible shielding. The factor $1/2$ in the objective accounts for the fact that each undirected shielded link is counted twice ($h_{ij} = h_{ji} = 1$). The LP relaxation of MILP (3.3) is tighter for smaller M, i.e., smaller M leads to a higher objective of the LP relaxation, which is a lower bound for the optimal shielding cost. The smallest M for a valid formulation in general is $M = F - 1$, which guarantees that any cut that contains the shielded link has capacity at least F.

There are exponential number of cut constraints (3.1). To reduce the number of constraints, consider a flow formulation which replaces the cut constraints by flow constraints (3.4), which guarantee F flow from s to d and have total number $|V|$. All

the variables and other constraints are identical with the cut formulation MILP (3.3).

$$\min \quad \sum_{(i,j)\in E} c_{ij}h_{ij}/2$$

$$\text{s.t.} \quad \sum_{\{j|(i,j)\in E\}} x_{ij} - \sum_{\{j|(j,i)\in E\}} x_{ji} = \begin{cases} F, \text{ if } i = s \\ -F, \text{ if } i = d \\ 0, \text{ otherwise} \end{cases} \tag{3.4}$$

$$x_{ij} \leq 1 + Mh_{ij} \quad \forall (i,j) \in E$$

$$h_{ij} - h_{ji} = 0 \quad \forall (i,j) \in E \tag{3.5}$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in E, n$$

$$h_{ij} = \{0,1\} \quad \forall (i,j) \in E \tag{3.6}$$

To further reduce the complexity of MILP (3.6), constraints (3.5) can be dropped and the factor $1/2$ in the objective can be removed accordingly. To see this, note that there exists an optimal solution where either x_{ij} or x_{ji} is zero for all (i,j). If both are nonzero in an optimal solution, the flow which has smaller value can be set to zero and subtracted from the other flow without violating any constraint. Therefore, either h_{ij} or h_{ij} is nonzero to guarantee that the link (i,j) is shielded and may carry up to $M+1$ units flow by dropping constraints (3.5), and the optimal shielding cost is given by $\sum_{(i,j)\in E} c_{ij}h_{ij}$. The formulation is stated as follows and contains only

$(|V| + |E|)$ constraints in addition to the boundary constraints.

$$\min \quad \sum_{(i,j)\in E} c_{ij} h_{ij}$$

$$\text{s.t.} \quad \sum_{\{j|(i,j)\in E\}} x_{ij} - \sum_{\{j|(j,i)\in E\}} x_{ji} = \begin{cases} F, \text{ if } i = s \\ -F, \text{ if } i = d \\ 0, \text{ otherwise} \end{cases} \tag{3.7}$$

$$x_{ij} \leq 1 + M h_{ij} \quad \forall (i,j) \in E \tag{3.8}$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in E, n$$

$$h_{ij} = \{0,1\} \quad \forall (i,j) \in E \tag{3.9}$$

Recall that we have an efficient combinatorial algorithm to eliminate the min cuts in Section 3.2.1. It is interesting to note that solving MILP (3.9) is efficient in order to eliminate the min cuts, where it suffices to let $M = 1$ in order to increase the effective min-cut by 1. Next we prove the following theorem.

Theorem 3. *MILP (3.9) is an integral formulation given $M = 1$ (i.e., the optimal solution of its LP relaxation is integral).*

Proof. A sufficient condition for integral formulation is that the constraint coefficient matrix is totally unimodular (TU), such that the feasible region polyhedron of the LP relaxation has integral extreme points. A matrix $A \in \mathbb{Z}^{m\times n}$ is TU if and only if $\forall I \subseteq \{1,2,\ldots,m\}$, $\exists I_1, I_2 \subseteq I$, $I_1 \cup I_2 = I$, $I_1 \cap I_2 = \emptyset$, such that [47]

$$|\sum_{i\in I_1} A_{ij} - \sum_{i\in I_2} A_{ij}| \leq 1, \ \forall j \in \{1,2,\ldots,n\}. \tag{3.10}$$

Each variable x_{ij} appears in two constraints in (3.7) with coefficients 1 and -1. Let A_1 denote the coefficient matrix of constraints (3.7). For any subset I^a of rows of A_1, $I_1 = I^a$, $I_2 = \emptyset$ clearly satisfies condition (3.10). Among the constraints (3.8), each variable x_{ij} and h_{ij} only appear in one constraint. Let A_2 denote the coefficient matrix of constraints (3.8) and let I^b denote any subset of the rows of A_2. Suppose

$I = I^a \cup I^b$ contains a row $u \in I^a$ and a row $v \in I^b$ that have nonzero coefficient of x_{ij}; then $v \in I_2$ if x_{ij} has the same coefficient as in $u \in I^a$, while $v \in I_1$ if x_{ij} has the opposite coefficient as in $u \in I^a$. The sums of coefficients of x_{ij} in I_1 and I_2 differ by at most 1. If two rows in I^a have nonzero coefficients of x_{ij}, these two rows are both in I_1 and the sum of the coefficients of x_{ij} is 0, and any row in I^b that has nonzero coefficient of x_{ij} can be either in I_1 or I_2, such that the sums of coefficients of x_{ij} in I_1 and I_2 differ by at most 1. Since h_{ij} only appears in one constraint, the sums of the coefficients of h_{ij} differ by at most 1 for all I_1, I_2. To conclude, condition (3.10) is satisfied and the constraint coefficient matrix is TU. □

Next we solve the above MILP to obtain the optimal shielding to eliminate small cuts between the SD pair Seattle-Miami in the XO network. To eliminate the min cuts, we obtain the same results as in Fig. 3-1. To eliminate cuts of size three or less, the optimal shielding is depicted in Fig. 3-5 with total cost 47.78. To eliminate cuts of size four or less, a path need to be shielded as in Fig. 3-6 with total cost 51.45. The running times to solve MILP (3.9) are smaller than 1 second to obtain the above results (executed on a desktop PC with Intel® Xeon®, 2.67 GHz CPU, 4 GB RAM and 64-bit operating system). The problem is easy to solve because only a single commodity flow need to be carried through the network between a single SD pair, and the number of constraints is small.

3.3 Increasing the effective min-cut of a network

In a network where each link has unit capacity, the min-cut of the network is the least number of link removals which separate the nodes into two disjoint sets. This is in contrast with the previous section that considers the min-cut of an SD pair, where the source and destination have to be in two disjoint sets. Consider the connection between the shielding problem and the augmentation problem. If the objective of shielding is to increase the effective min-cut from $k-1$ to k, shielding a link can be viewed as augmenting *one* parallel link, which suffices to carry one more unit of flow. Therefore, the problem of increasing the effective min-cut by one (from $k-1$ to k) is

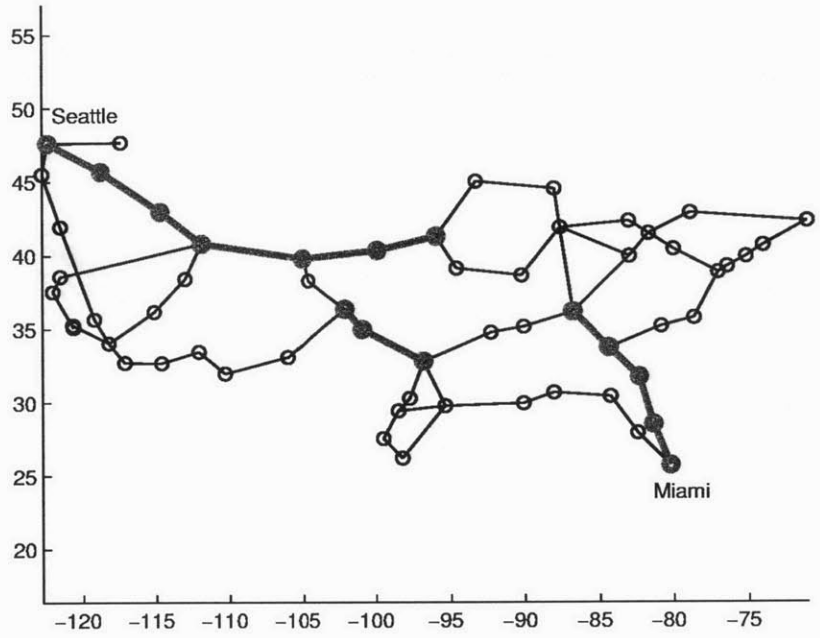

Figure 3-5: Optimal shielding to eliminate cuts of size $|C| \leq 3$ between an SD pair in the XO network.

equivalent to the augmentation problem that guarantees a k connected graph, which has been proved to be NP-hard [22].

To increase the effective min-cut and eliminate small cuts of a network, considering the SD pairs separately may not lead to the optimal solution because of the coupling between the SD pairs. Therefore, the algorithm that eliminates small cuts of a single SD pair is not easily extended to eliminate small cuts of the network. In this section we develop algorithms to eliminate small cuts of the network.

3.3.1 Eliminating min cuts of a network

First consider a special case where only one SD pair is the bottleneck for the network. The Gomory-Hu algorithm calculates network min-cut by solving $|V| - 1$ min-cut problems for different SD pairs [23]. The result is represented by a Gomory-Hu tree where the link between two nodes have a weight that is equal to the min-cut between

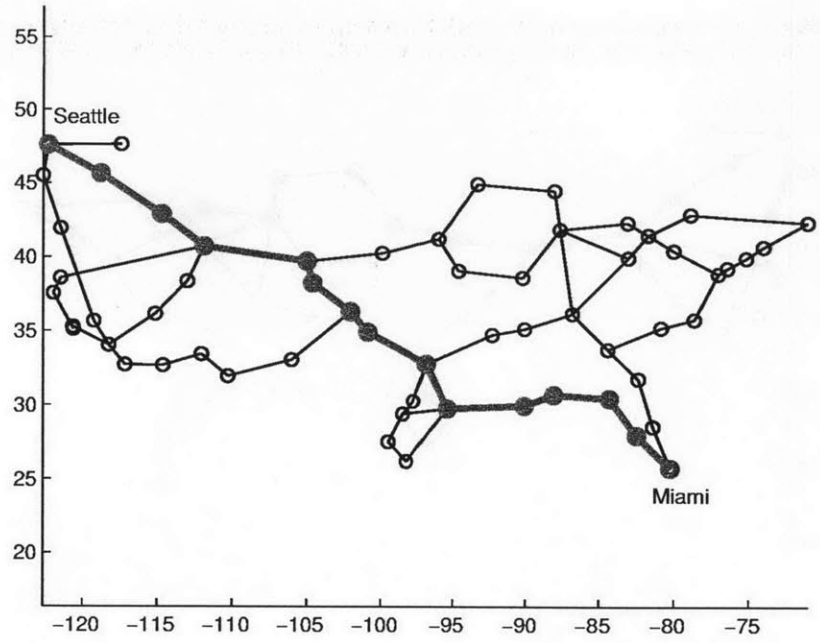

Figure 3-6: Optimal shielding to eliminate cuts of size $|C| \leq 4$ between an SD pair in the XO network.

these two nodes. If the weight between an SD pair is smaller than all the others, eliminating the min cut of this SD pair also eliminates the min cut of the network. However, it is possible that more than one SD pair has the minimum weight and eliminating the min cut separately for each SD pair may not be the optimal solution for the network.

Next we develop an algorithm that provides sufficient shielding to eliminate all the min cuts of the network. Any cut that separates the network has to contain at least one link in a spanning tree. Otherwise, all the nodes are connected by the spanning tree and the removed links do not form a cut. It is natural to expect that shielding the links belonging to the min cuts in a spanning tree eliminates all the min cuts of the network. Next we formally prove the theorem.

Theorem 4. *Given any spanning tree, shielding all the links in that spanning tree that belong to min cuts is sufficient to eliminate all the min cuts of the network.*

Proof. The proof is similar to the sufficiency proof in Theorem 1. Let m be the min-cut of the network. Eliminating all the min cuts is equivalent to guaranteeing that the network is connected after m unshielded links' removals. Let C denote the set of links belonging to min cuts. There does not exist a set of m link removals that disconnects the network and contains any link not in C. Shielding the links in C in any spanning tree guarantees a spanning tree in the network after m link removals. In other words, it is sufficient to guarantee that the network is connected after m link removals and thus eliminates all the min cuts of the network. □

However, shielding all the links in a spanning tree that belong to min cuts is not necessary in order to eliminate all the min cuts. Consider the clique of four nodes as in Fig. 3-7. The min-cut is three in this clique and all the links belong to min cuts. Instead of shielding a spanning tree, only two diagonal links need to be shielded in order to eliminate all the min cuts and increase the effective min-cut by one.

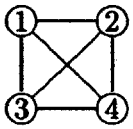

Figure 3-7: Shielding the two diagonal links increases the network effective min-cut by one.

3.3.2 Eliminating small cuts of a network

We extend MILP (3.6) to formulate the shielding problem to eliminate all the cuts with size smaller than F that disconnect the network. The only difference compared with MILP (3.6) is that the max-flow between every SD pair should be at least F, guaranteed by constraints (3.11). Note that the number of constraints cannot be further reduced as in MILP (3.9), because flows between difference SD pairs may travel in different directions of a link and $h_{ij}h_{ji}$ may not be always 0 by dropping

constraints (3.12).

$$\min \quad \sum_{ij \in E} c_{ij} h_{ij}/2$$

$$\text{s.t.} \quad \sum_{\{j|(i,j)\in E\}} x_{ij}^{(sd)} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(sd)} = \begin{cases} F, \text{ if } i = s \\ -F, \text{ if } i = d \ \forall s, d \\ 0, \text{ otherwise} \end{cases} \tag{3.11}$$

$$x_{ij}^{(sd)} \leq 1 + M h_{ij} \ \ \forall (i,j) \in E$$

$$h_{ij} - h_{ji} = 0 \ \ \forall (i,j) \in E \tag{3.12}$$

$$x_{ij}^{(sd)} \geq 0 \ \ \forall (i,j) \in E, n$$

$$h_{ij} = \{0,1\} \ \ \forall (i,j) \in E \tag{3.13}$$

3.4 Numerical results

We first solve MILP (3.13) to obtain the optimal shielding to increase the effective min-cut of the XO network. Table 3.1 shows that the running time increases significantly as the effective min-cut increases. This is consistent with our analysis that the formulation is stronger for smaller values of effective min-cut. The optimal shielding that increases the effective min-cut to 3 is depicted in Fig. 3-8. From the figure we observe that many links need to be shielded if the min-cuts between many adjacent SD pairs are smaller than the required effective min-cut.

Effective min-cut	2	3	4	5
Cost	4.92037	134.076	147.741	163.39
Time (s)	0.13	10.29	70.08	1143.68

Table 3.1: Increasing the effective min-cut of the XO network.

Table 3.2 compares the running times of solving the MILP in a 150 node network to satisfy different effective min-cuts. The network is a connected Erdos-Renyi random graph with average degree 7. It can be observed that the running time also increases significantly as the effective min-cut increases.

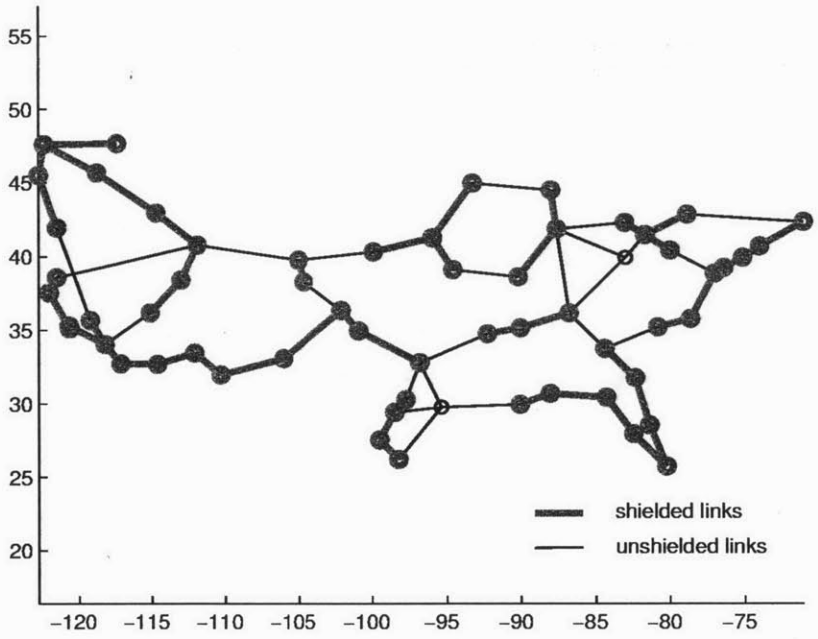

Figure 3-8: Optimal shielding to increase the effective min-cut to 3 in the XO network.

Effective min-cut	3	4	5	6	7	8
Time (s)	2.85618	4.54428	9.6766	15.0489	166.282	972.221

Table 3.2: Running time comparisons for guaranteeing different effective min-cuts in a 150 node random graph.

3.5 Conclusion

In this chapter we developed algorithms to increase the effective min-cut by shielding, which is key in designing robust networks that can tolerate random failures. We developed an efficient algorithm to optimally eliminate the min cuts and increase the effective min-cut by one for a single SD pair. To increase the effective min-cut by an arbitrary value, we developed a MILP to obtain the optimal solutions, and reduced the number of constraints by exploiting problem properties to solve realistic size problems. To eliminate the min cuts and increase the effective min-cut by one for the entire network, we developed an algorithm to provide sufficient shielding and obtained an upper bound on the optimal shielding cost. Finally we extended the

MILP to obtain the optimal shielding to increase the effective min-cut of the entire network by an arbitrary value.

Chapter 4

Shielding critical nodes in random graphs

In Chapter 2 we developed optimal shielding algorithms to guarantee network connectivity after any failure. While the algorithms can be used to shield backbones, the Internet is far more than backbones, with over two billion Internet users around the world. The entire network is too large to be considered globally for shielding optimization. Moreover, it is unnecessary to guarantee the connectivity of the entire network at expensive cost, since many users have flexible quality of service requirements and can tolerate temporary Internet malfunction. An Internet service provider may instead consider guaranteeing a large fraction of users to be able to use Internet service in case of failures, and then recover service for the affected users by restoration. To consider which parts of the network need to be shielded in advance to ensure that a large fraction of users are connected, a good starting point is to study random graphs, which possess nice analytical properties.

There is rich literature that discusses the resilience of the random graphs to random failures and intentional attacks under various network configurations [26, 3, 24]. These works study the effect of node and link removals on cluster size, average path length, and other measures of robustness. For intentional removals, nodes are removed in descending order of centrality measures such as degree or betweenness [11, 24]. These removal strategies are considered as heuristics without rigorous theoretical jus-

tifications, except the work in [7] that analytically studies the effect of removing early stage nodes in preferential attachment model.

Few papers consider the effects of shielding in random graphs, except the works [49, 28]. Power law network's robustness can be significantly improved by even imperfect protection of high degree nodes [49]. However, it is unclear whether the high degree nodes are indeed the most important nodes to be shielded.

We aim to determine which nodes are important to shield in configuration model, where analytical tools are available in calculating the size of the giant component [13, 37]. We analytically justify the importance of shielding the high degree nodes on graph connectivity. In power law networks ($p_k = ck^{-\alpha},\ k = m, \ldots, K$) where the exponent $\alpha < 3$, shielding a small fraction of nodes suffices to guarantee the existence of a giant component in the asymptotic case ($K \to \infty$) even if all the unshielded nodes are removed.

4.1 Configuration model and transformation method

Due to analytical tractability, configuration model is often used to generate random graphs with given node degree distribution [38]. Given a set of nodes and their degrees, each node is attached by several stubs, whose number equals to the node degree. Links are generated by a random matching of stubs. Note that it is possible to have multi-edges and self loops. However, if the number of nodes is very large, the fraction of multi-edges and self loops is negligible.

To study random graphs constructed under the configuration model, transformation method is widely used [37, 18, 17]. Since each link randomly matches two stubs, the number of nodes that can eventually be reached by following the end of each link is statistically identical. This is the key condition that the transformation method relies on, which facilitates the calculation of probability generating functions for cluster sizes. To follow the convention, we use the term generating functions to refer to probability generating functions. In the remaining of this section, we summarize the methods in [37] of calculating the size and existence condition of the giant component

using the transformation method.

Given node degree distribution $\{p_k\}$, its generating function is:

$$G_0(x) = \sum_k p_k x^k. \tag{4.1}$$

Following a randomly chosen link, the probability that a degree k node is reached is $kp_k / \sum_k kp_k$. The generating function for the degree of a node reached by a randomly chosen link is:

$$\frac{\sum_k kp_k x^k}{\sum_k kp_k} = x\frac{G_0'(x)}{G_0'(1)}. \tag{4.2}$$

Let r_i be the probability that the node reached by a randomly chosen link has i additional links (the node has degree $i+1$), i.e., $r_i = (i+1)p_{i+1} / \sum_k kp_k$. To characterize the distribution of the number of additional links incident to the node reached by a randomly chosen link, the following generating function is used:

$$G_1(x) = \frac{\sum_k kp_k x^{k-1}}{\sum_k kp_k} = \frac{G_0'(x)}{G_0'(1)} = \frac{1}{x}G_0'(x). \tag{4.3}$$

Let $H_1(x)$ be the generating function for the total number of nodes that can eventually be reached by a randomly chosen link. With probability r_i, a randomly chosen link reaches a node that has i additional links, in which case the generating function for the total number of nodes reached by the link is $xH_1^i(x)$ (the additional x accounts for the node which is attached to the link). The generating function $H_1(x)$ can be obtained by solving the following equation.

$$H_1(x) = xr_0 + xr_1H_1(x) + xr_2[H_1(x)]^2 + \ldots = xG_1(H_1(x)). \tag{4.4}$$

Finally consider the cluster size distribution. Given a randomly chosen node, the generating function for its degree (i.e., the number of incident links) is $G_0(x)$. For each link, $H_1(x)$ is the generating function for the number of nodes that can eventually be reached. Therefore, the generating function for the size of the cluster that contains a

randomly chosen node is:

$$H_0(x) = xG_0(H_1(x)). \tag{4.5}$$

In the asymptotic case where the number of nodes approaches infinity, the probability that a randomly chosen node is in any finite size component is $H_0(1)$. The size of the giant component is approximated by $1 - H_0(1)$. If $H_0(1) < 1$, there exists a giant component, which corresponds to the well known condition [37, 18, 17]

$$\sum_k k(k-2)p_k > 0.$$

Next we summarize the results in [37] that use the transformation method to study the effect of removing nodes (and all the links incident to the removed nodes) on random graphs constructed under the configuration model. Suppose that the probability that a degree k node is occupied (i.e., not removed) is q_k. Thus, the fraction of occupied nodes with degree k is $p_k q_k$. The generating function for the node degree, in analogy of $G_0(x)$, is:

$$F_0(x) = \sum_k p_k q_k x^k.$$

The generating function for the number of extra links of an occupied node reached by a randomly chosen link, in analogy of $G_1(x)$, is:

$$F_1(x) = \frac{\sum_k k p_k q_k x^{k-1}}{\sum_k k p_k}.$$

The generating function for the number of nodes eventually reached by a randomly chosen link is $H_1(x)$, calculated by the following equation. If the node reached by the link is unoccupied (with probability $1 - F_1(1)$), the total number of nodes eventually reached is 0 and contributes the term $(1 - F_1(1))x^0 = 1 - F_1(1)$.

$$H_1(x) = 1 - F_1(1) + xF_1(H_1(x)). \tag{4.6}$$

Finally, the generating function for the size of the component containing a ran-

domly chosen node is as follows. The probability that a randomly chosen node is unoccupied is $1 - F_0(1)$, contributing to the first term of the right hand side of the equation.

$$H_0(x) = 1 - F_0(1) + xF_0(H_1(x)).$$

In the asymptotic case where the number of nodes approached infinity, the fraction of nodes in the giant component is approximated by

$$S = 1 - H_0(1) = F_0(1) - F_0(H_1(1)), \tag{4.7}$$

where $H_1(1)$ is given by Eq. (4.6).

4.2 Importance of high degree nodes on connectivity

Based on the transformation method, the authors calculated the giant component size after node removals and observed that power law networks are robust to random failures but vulnerable to intentional attacks [18, 17]. We first analytically demonstrate that the high degree nodes are more important than low degree nodes for connectivity in random graphs constructed under the configuration model.

To obtain the size of the giant component using Eq. (4.7), $H_1(x)$ needs to be first calculated using Eq. (4.6). Let $u = H_1(1)$, Eq. (4.6) can be written as

$$u = 1 - F_1(1) + F_1(u).$$

Equivalently,

$$1 - u = \frac{\sum_k k p_k q_k (1 - u^{k-1})}{\sum_k k p_k}. \tag{4.8}$$

After obtaining u from Eq. (4.8), the size of the giant component is given by:

$$S = \sum_k p_k q_k - \sum_k p_k q_k u^k. \tag{4.9}$$

Nodes are critical to graph connectivity if their removals lead to a sharp decrease

of the size of the giant component. In order to determine the critical nodes, we solve the following optimization problem with decision variables q_k and u, which minimizes the size of the giant component by removing δ fraction of nodes. Constraint (4.10) is equivalent to Eq. (4.8), and constraint (4.11) guarantees that ϵ fraction of nodes are removed.

$$\begin{aligned}
\min \quad & f = \sum_k p_k q_k - \sum_k p_k q_k u^k \\
\text{s.t.} \quad & \sum_{k\geq 2} k p_k q_k (1 + u + u^2 + \cdots + u^{k-2}) = \sum_k k p_k \qquad (4.10) \\
& \sum_k p_k (1 - q_k) = \delta \qquad (4.11) \\
& 0 \leq q_k \leq 1 \quad \forall k \\
& 0 \leq u \leq 1
\end{aligned}$$

In the following we provide a solution to the optimization problem. The key is to show

$$f(q_1, q_2, \ldots, q_i, \ldots, q_j, \ldots, q_{n-1}, q_n) < f(q'_1, q'_2, \ldots, q'_i, \ldots, q'_j, \ldots, q'_{n-1}, q'_n), \qquad (4.12)$$

in condition that

$$q_i > q'_i,\ q_j < q'_j,\ p_i q_i + p_j q_j = p_i q'_i + p_j q'_j,\ i < j$$

$$q_k = q'_k \quad \forall k \neq i, j.$$

Constraint (4.10) determines the value of u. Since $ip_i(q_i - q'_i) < jp_j(q'_j - q_j)$, it is easy to see that $\sum_{k\geq 2} k p_k q_k < \sum_{k\geq 2} k p_k q'_k$, and therefore $u > u'$ in order to satisfy constraint (4.10). We are concerned with the case where there is a giant component where $u < 1$. Even for fixed u, $\sum_k p_k q_k u^k > \sum_k p_k q'_k u^k$, since $u^i > u^j$ and $p_i(q_i - q'_i) = p_j(q'_j - q_j) > 0$ for $i < j$. Given that $u > u'$, $\sum_k p_k q_k u^k > \sum_k p_k q'_k u'^k$, and the inequality (4.12) holds. In other words, the occupancy of high degree nodes results in a larger giant component compared with the occupancy of the same number

of low degree nodes. The optimal solution to minimize the size of the giant component is to set q_k as small as possible for large k while satisfying constraint (4.11).

Using the same argument, it is clear that shielding the highest degree nodes maximizes the size of the giant component, since it avoids the highest degree nodes being removed.

Next we consider the phase transition of the giant component. Similar to (4.1), the critical condition for the existence of the giant component is $\sum_k kp_k \leq \sum_{k\geq 2} kp_k q_k(k-1)$ [13]. For power law random graphs ($p_k = ck^{-\alpha}$, $k = m, \ldots, K$), shielding a small fraction of highest degree nodes guarantees a giant component if $\alpha < 3$ in the asymptotic case where $K \to \infty$. To prove the result, we use continuous approximation as in [17]. Suppose that all the nodes with degree K_s or above are shielded, accounting for ϵ fraction of total nodes ($\int_{K_s}^{K} p_k dk = \epsilon$), while the nodes with degree K_0 or above (except the shielded nodes) are removed ($\int_{K_0}^{K_s} p_k dk = \delta$). We next prove that the relation

$$\int_m^K kck^{-\alpha}dk \leq \int_m^{K_0} k(k-1)ck^{-\alpha}dk + \int_{K_s}^{K} k(k-1)ck^{-\alpha}dk$$

holds for $m < K_0 < K_s \ll K$, in condition that $\alpha < 3$. Note that even a small value of ϵ may guarantee that $K_s \ll K$, since $K \to \infty$ and the number of very high degree nodes is small.

Proof. After integration, the inequality that we aim to prove becomes:

$$\begin{aligned}\frac{1}{2-\alpha}(K^{2-\alpha} - m^{2-\alpha}) \quad &\leq \frac{1}{3-\alpha}(K^{3-\alpha} - K_s^{3-\alpha} + K_0^{3-\alpha} - m^{3-\alpha}) \\ &\quad -\frac{1}{2-\alpha}(K^{2-\alpha} - K_s^{2-\alpha} + K_0^{2-\alpha} - m^{2-\alpha}). \qquad (4.13)\end{aligned}$$

We prove (4.13) in the following two ranges of α.

1. $\alpha < 2$

Given $3-\alpha > 2-\alpha > 0$, $i^\beta < j^\beta$ for $i < j, \beta > 0$, it suffices to show that

$$\frac{K^{3-\alpha} - K_s^{3-\alpha} + K_0^{3-\alpha} - m^{3-\alpha}}{K^{2-\alpha} - m^{2-\alpha} + K^{2-\alpha} - K_s^{2-\alpha} + K_0^{2-\alpha} - m^{2-\alpha}} \geq \frac{3-\alpha}{2-\alpha}.$$

Consider the asymptotic case where $K \to \infty$.

$$\lim_{K\to\infty} \frac{K^{3-\alpha} - K_s^{3-\alpha} + K_0^{3-\alpha} - m^{3-\alpha}}{K^{2-\alpha} - m^{2-\alpha} + K^{2-\alpha} - K_s^{2-\alpha} + K_0^{2-\alpha} - m^{2-\alpha}} = K > \frac{3-\alpha}{2-\alpha}.$$

2. $2 < \alpha < 3$

Since $2 - \alpha < 0$, the following inequalities hold: $0 < m^{2-\alpha} - K^{2-\alpha} < m^{2-\alpha}$, $0 < m^{2-\alpha} - K_0^{2-\alpha} < m^{2-\alpha}$, and $0 < K_s^{2-\alpha} - K^{2-\alpha} < K_s^{2-\alpha}$.

$$\lim_{K\to\infty} -\frac{K^{3-\alpha} - K_s^{3-\alpha} + K_0^{3-\alpha} - m^{3-\alpha}}{K^{2-\alpha} - m^{2-\alpha} + K^{2-\alpha} - K_s^{2-\alpha} + K_0^{2-\alpha} - m^{2-\alpha}} \geq \frac{K^{3-\alpha} - K_s^{3-\alpha}}{3m^{2-\alpha}} \geq \frac{3-\alpha}{\alpha-2}.$$

□

We briefly discuss the region $\alpha > 3$. This region corresponds to the case where there are many nodes with small degrees and the graph is loosely connected.

Note that $i^\beta > j^\beta$ for $i < j$ and $\beta < 0$.

$$\lim_{K\to\infty} \frac{m^{3-\alpha} - K_0^{3-\alpha} + K_s^{3-\alpha} - K^{3-\alpha}}{2m^{2-\alpha} - 2K^{2-\alpha} - K_0^{2-\alpha} + K_s^{2-\alpha}} \geq \frac{m^{3-\alpha} - K_0^{3-\alpha}}{2m^{2-\alpha}}. \tag{4.14}$$

It is reasonable to assume that $m = 1$ since there exist degree 1 nodes in power law random graphs. If there is no node removals, $K_0 = K$ and the condition for the existence of the giant component is $\alpha < 4$, by solving the equation $\frac{m^{3-\alpha}}{2m^{2-\alpha}} > \frac{\alpha-3}{\alpha-2}$ [1]. If a large number of nodes are removed, K_0 is close to m and there may not exist a giant component.

4.3 Numerical results

We provide numerical results to show the importance of high degree nodes in random graphs with node degrees following the power law distribution $p_k = ck^{-\alpha}$ ($k = 1, \ldots, 100$). By removing the selected 2% nodes, the fraction of nodes in the giant component is depicted in Fig. 4-1. The 2% removed nodes are selected to be in the

[1] Strictly speaking, there is no giant component if $\alpha > 3.48$ by exactly calculating the sum in condition (4.1) [2]. However, we use the continuous approximation to approximate the sum by integral, to be consistent with the analysis in this chapter and previous literature [18, 17].

degree intervals in deceasing order, i.e., the i-th pair corresponds to the case where nodes are removed whose degrees are in the range $(100 - 2i)\% \sim (102 - 2i)\%$. The solid line is obtained by calculating S using Eq. 4.9. The symbols are simulation results averaged over 10 instances of random graphs with 10^4 nodes, with error bars showing the 95% confidence intervals based on Student's t distribution. It can be observed that the higher degree nodes removed, the smaller the giant component size. Removing the highest 2% degree nodes is possible to break down the giant component, which is in consistent with previous result that power law network is fragile to intentional removals. We also obtained numerical results for random graphs where the node degrees follow Poisson distribution $p_k = e^{-\lambda}\lambda^k/k!$, and observed that removing higher degree nodes still results in smaller size of the giant component in Fig. 4-2. However, the differences are smaller, because the degree range for most nodes are limited in Poisson random graphs, whereas there are more nodes with large degrees in power law random graphs.

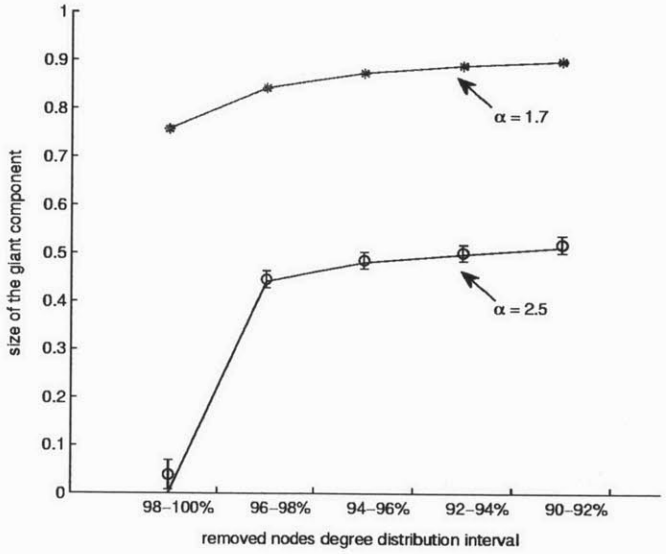

Figure 4-1: Size of the giant component after removing the selected 2% nodes in power law random graphs.

Then we consider the effect of shielding in case of intentional node removals. The fraction of nodes in the giant component is calculated after shielding 2% nodes and

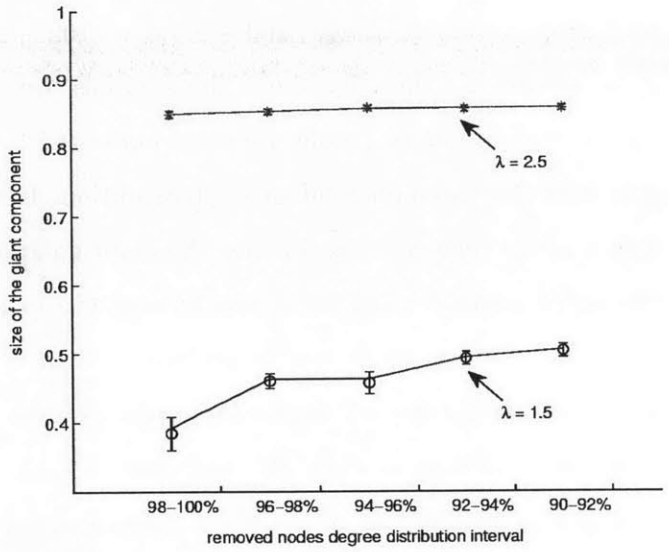

Figure 4-2: Size of the giant component after removing the selected 2% nodes in Poisson random graphs.

removing the remaining 10% highest degree nodes. We still use the solid line to represent the analytical results, and use symbols with error bars to represent simulation results of 10 instances of random graphs. As Fig. 4-3 suggests, shielding higher degree nodes leads to a larger giant component, and shielding the highest 2% degree nodes suffices to guarantee the existence of a giant component in power law random graphs. However, in Poisson random graphs, shielding the highest degree nodes does not necessarily guarantee the existence of a giant component. For example, if $\lambda = 1.5$, there is no giant component after shielding the top 2% nodes while removing the remaining 10% highest degree nodes.

Next we consider the effect of shielding under random node removals. Although the nodes in power law random graphs have a wide range of degrees, given that power law random graphs are robust to random node removals and the decrease of the size of the giant component is small even without shielding, shielding higher degree nodes marginally increases the size of the giant component, depicted in Fig. 4-4. In Poisson random graphs, the range of degrees is limited. Shielding higher degree nodes marginally increases the size of the giant component since the top 2% highest degree

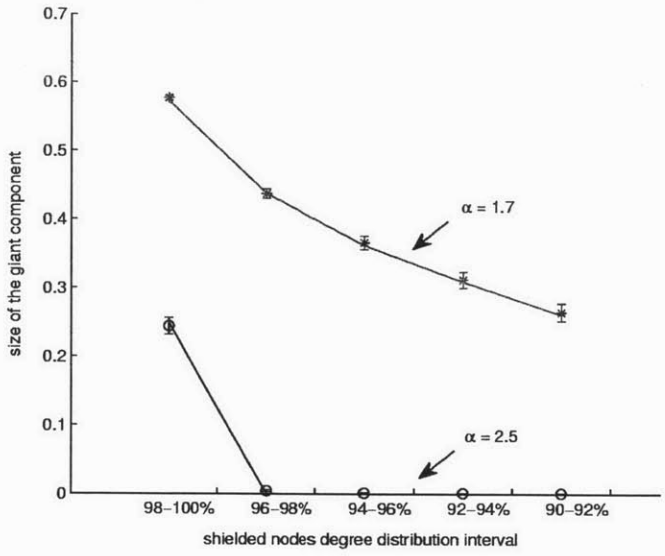

Figure 4-3: Size of the giant component after shielding the selected 2% nodes and removing 10% unshielded highest degree nodes in power law random graphs.

nodes have similar degrees compared with the nodes whose degrees are in the range $90\% \sim 92\%$.

Finally we consider the transition point for the existence of the giant component after shielding the highest degree nodes. After shielding only 1% highest degree nodes, there always exists a giant component in power law random graphs if $\alpha < 3$, as depicted in Fig. 4-5. However, in Poisson random graphs, shielding 1% highest degree nodes does not guarantee the existence of a giant component as depicted in Fig. 4-6. This can be explained intuitively by the fact that in power law random graphs, links attached to even a small fraction of the highest degree nodes account for a significant portion among the entire links, and these nodes are likely to be connected with one another to form part of the giant component. However, since the degree range is limited in Poisson random graphs, the links attached to a small fraction of highest degree nodes do not account for a significant portion and therefore these nodes are not likely connected after removing the other nodes and links.

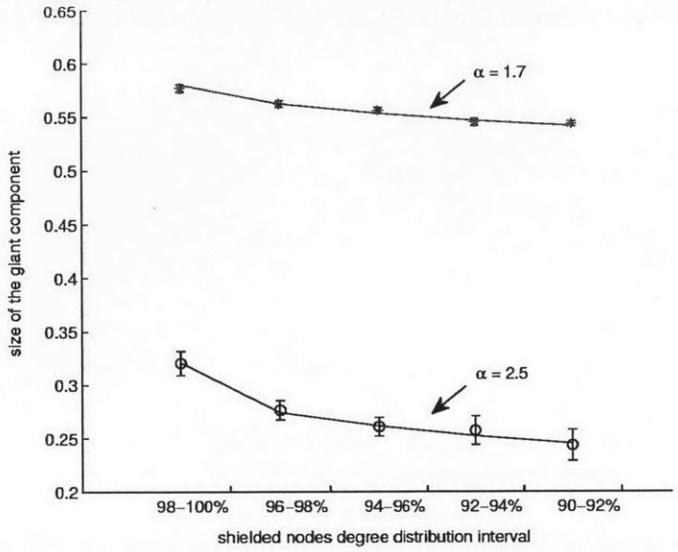

Figure 4-4: Size of the giant component after shielding the selected 2% nodes and randomly removing 30% unshielded nodes in power law random graphs.

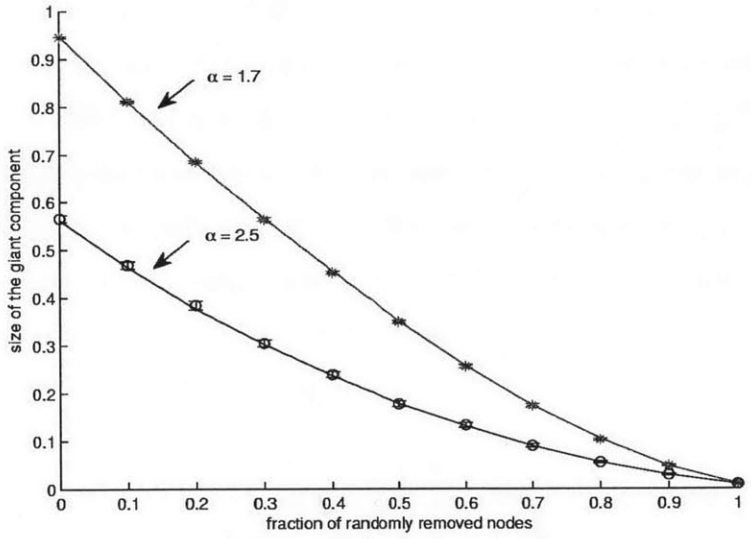

Figure 4-5: Size of the giant component after shielding the selected 1% nodes vs. the fraction of randomly removed nodes in power law random graphs.

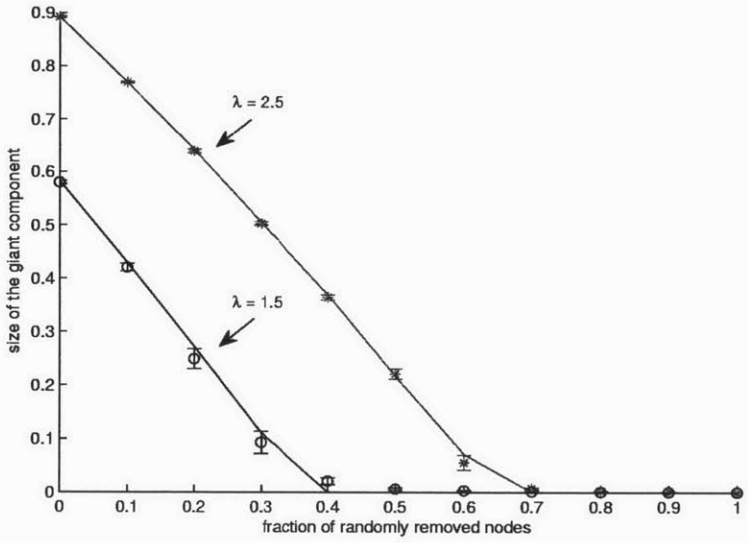

Figure 4-6: Size of the giant component after shielding the selected 1% nodes vs. the fraction of randomly removed nodes in Poisson random graphs.

Chapter 5

Conclusion and future work

In this thesis we considered shielding to enhance network robustness. Shielding critical parts of the network provides an alternative approach of designing robust networks, in addition to redundancy provisioning which has been widely studied in the previous literature. We developed algorithms to shield critical parts of networks under several realistic failure scenarios.

Under the geographical and general failure models, we determined the minimum cost shielding to guarantee that the network is connected after any possible failure. We used decomposition techniques to solve network problems that have realistic sizes. Besides, we observed that significantly less shielding cost is required if the connectivity requirement is slightly relaxed. Under the random failure model, we developed algorithms to increase network effective min-cut by shielding critical links, which is the key in improving network robustness especially when the failure probability is small. Finally, we considered shielding in random graphs and demonstrated that high degree nodes are important in the connectivity of random graphs constructed under the configuration model. In power law random graphs, shielding a small fraction of nodes guarantees the existence of a giant component if the power law exponent is less than three.

The algorithms in this thesis can be extended to solve several classes of network design problems related to shielding. One class of network design problem considers guaranteeing network k connectivity ($k \geq 2$). Since shielded links may still fail in case

of disasters or attacks, it is reasonable to consider the optimal shielding to guarantee that the network is still connected even after one (or more) shielded link's failure. For example, under the geographical failure model, shielded links in each failure region need to form a two (or higher) connected graph, after contracting the nodes outside the failure region. To solve this problem, the MILP formulations in Chapter 2 can be extended to guarantee two or higher connectivity with significantly larger number of constraints. Future work may consider developing efficient algorithms to solve these more general connectivity problems.

Another direction of future work is to consider the effects of probabilistic survivability of shielded links. For example, the probability that each link survives a failure depends on its shielding level and cost. In this case, an important problem is to maximize network reliability (under certain failure models) given the shielding budget constraint. The difficulty of this problem is that the objective function and constraints of the optimization problem may not be convex. A promising approach is to use the augmented Lagrangian method to solve this reliability optimization problem [42, 6]. The augmented Lagrangian method convexifies the objective function along certain directions near the boundary of the feasible region by adding a quadratic penalty function, which makes it easier to find a global minimum near the boundary of the feasible region using convex optimization techniques. This is particularly helpful in the reliability optimization problem, since the optimal shielding tends to use up the available budget and the optimal solution is close to the boundary of the feasible region.

To maximize network reliability under random link failure models, in addition to increasing the effective min-cut, it is also desirable to minimizing the number of cuts in the case where the shielding budget is insufficient to eliminate all the cuts of given size. Although this can be viewed as a variation of hitting set problem where each set is a cut and each element is a link, the direct approach of solving the hitting set problem is intractable given the large number of cuts. Future work may consider developing efficient algorithms to obtain near optimal solutions based on the recent advancement of solutions to implicit hitting set problem [15].

Reliable communication becomes more important as the communication networks increasingly support people's work and operations. In summary, the algorithms in this thesis can be used to design robust networks of realistic sizes that can survive intentional attacks or random failures, and can be extended to satisfy more stringent design and reliability requirements.

Bibliography

[1] P.K. Agarwal, A. Efrat, S.K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. The resilience of WDM networks to probabilistic geographical failures. *IEEE/ACM Trans. Netw.*, 21(5):1525–1538, Oct 2013.

[2] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10(1):53–66, 2001.

[3] R. Albert, H. Jeong, and A.L. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[4] D. Bertsimas and J. Tsitsiklis. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[5] D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific series in optimization and neural computation. Athena Scientific, 1997.

[6] E. G. Birgin and J. M. Martínez. Improving ultimate convergence of an augmented Lagrangian method. *Optimization Methods Software*, 23(2):177–195, April 2008.

[7] Bela Bollobas and Oliver Riordan. Robustness and vulnerability of scale-free random graphs. *Internet Mathematics*, 1(1):1–35, 2003.

[8] Booz Allen Hamilton Inc. Electromagnetic pulse survivability of telecommunication assets. Feb 1987.

[9] J. Borland. Analyzing the Internet collapse. *MIT Technology Review*, February 2008.

[10] Sylvie Borne, Eric Gourdin, Bernard Liau, and A.Ridha Mahjoub. Design of survivable IP-over-optical networks. *Annals of Operations Research*, 146(1):41–73, 2006.

[11] Ulrik Brandes and Daniel Fleischer. Centrality measures based on current flow. In *Proceedings of the 22Nd Annual Conference on Theoretical Aspects of Computer Science*, STACS'05, pages 533–544, Berlin, Heidelberg, 2005. Springer-Verlag.

[12] Gerald Brown, Matthew Carlyle, Javier Salmerón, and Kevin Wood. Defending critical infrastructure. *Interfaces*, 36(6):530–544, November 2006.

[13] Duncan S. Callaway, M. E. J. Newman, Steven H. Strogatz, and Duncan J. Watts. Network robustness and fragility: Percolation on random graphs. *Phys. Rev. Lett.*, 85:5468–5471, Dec 2000.

[14] C. Cao, M. Zukerman, W. Wu, J. H. Manton, and B. Moran. Survivable topology design of submarine networks. *IEEE/OSA J. Lightwave Technol.*, 31(5):715–730, March 2013.

[15] Karthekeyan Chandrasekaran, Richard Karp, Erick Moreno-Centeno, and Santosh Vempala. Algorithms for implicit hitting set problems. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 614–629. SIAM, 2011.

[16] Richard L. Church, Maria P. Scaparra, and Richard S. Middleton. Identifying critical infrastructure: The median and covering facility interdiction problems. *Annals of the Association of American Geographers*, 94(3):pp. 491–502.

[17] Reuven Cohen, Keren Erez, Daniel B. Avraham, and Shlomo Havlin. Breakdown of the Internet under intentional attack. *Physical Review Letters*, 86(16):3682–3685, April 2001.

[18] Reuven Cohen, Keren Erez, Daniel ben Avraham, and Shlomo Havlin. Resilience of the Internet to random breakdowns. *Phys. Rev. Lett.*, 85:4626–4628, Nov 2000.

[19] William H. Cunningham. Optimal attack and reinforcement of a network. *J. ACM*, 32(3):549–561, July 1985.

[20] Marcin Dziubinski and Sanjeev Goyal. Network design and defence. *Games and Economic Behavior*, 79(0):30 – 43, 2013.

[21] J. S. Foster, E. Gjelde, W. R. Graham, R. J. Hermann, H. M. Kluepfel, R. L. Lawson, G. K. Soper, L. L. Wood, and J. B. Woodard. Report of the commission to assess the threat to the united states from electromagnetic pulse (EMP) attack, critical national infrastructures. April 2008.

[22] András Frank. Augmenting graphs to meet edge-connectivity requirements. *SIAM J. Discret. Math.*, 5(1):25–53, February 1992.

[23] Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *Journal of the Society for Industrial & Applied Mathematics*, 9(4):551–570, 1961.

[24] Jean-Loup Guillaume, Matthieu Latapy, and Clémence Magnien. Comparison of failures and attacks on random and scale-free networks. In Teruo Higashino, editor, *OPODIS*, volume 3544 of *Lecture Notes in Computer Science*, pages 186–196. Springer, 2004.

[25] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13:311–329, 1988.

[26] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack vulnerability of complex networks. *Phys. Rev. E*, 65(5):056109, May 2002.

[27] Hervé Kerivin and A. Ridha Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, August 2005.

[28] Minkyu Kim and M. Medard. Robustness in large-scale random networks. In *INFOCOM 2004*, volume 4, pages 2364–2373 vol.4, March 2004.

[29] Way Kuo and V.R. Prasad. An annotated overview of system-reliability optimization. *IEEE Transactions on Reliability*, 49(2):176–187, Jun 2000.

[30] Duan Li, Xiaoling Sun, and Ken McKinnon. An exact solution method for reliability optimization in complex systems. *Annals of Operation Research*, 133:129–148, 2005.

[31] Changzheng Liu, Yueyue Fan, and Fernando Ordóñez. A two-stage stochastic programming model for transportation network protection. *Computers and Operations Research*, 36(5):1582 – 1590, 2009.

[32] Guido Maier, Achille Pattavina, Simone De Patre, and Mario Martinelli. Optical network survivability: Protection techniques in the WDM layer. In *Photonic Networks Communications*, pages 251–269, 2002.

[33] M. Medard, D. Marquis, R.A. Barry, and S.G. Finn. Security issues in all-optical networks. *IEEE Network*, 11(3):42–48, May 1997.

[34] M. Medard, D. Marquis, R.A. Barry, and S.G. Finn. Security issues in all-optical networks. *IEEE Network*, 11(3):42–48, May 1997.

[35] Sebastian Neumayer, Alon Efrat, and Eytan Modiano. Geographic max-flow and min-cut under a circular disk failure model. In *INFOCOM*, pages 2736–2740, 2012.

[36] Sebastian Neumayer, Gil Zussman, Reuven Cohen, and Eytan Modiano. Assessing the vulnerability of the fiber infrastructure to disasters. *IEEE/ACM Trans. Netw.*, 19(6):1610–1623, December 2011.

[37] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.

[38] Mark Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[39] D. Papadimitriou et al. Inference of shared risk link groups. Internet-Draft draft-many-inference-srlg-02.txt, IETF Secretariat, November 2001.

[40] Michal Pióro and Deepankar Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Publishers Inc., 2004.

[41] J. Plesnik. Equivalence between the minimum covering problem and the maximum matching problem. *Discrete Mathematics*, 49(3):315 – 317, 1984.

[42] R. Tyrrell Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM Journal on Control*, 12(2), 1974.

[43] Lawrence V. Snyder, Maria P. Scaparra, Mark S. Daskin, and Richard L. Church. Planning for disruptions in supply chain networks. In *Tutorials in Operations Research. INFORMS*, 2006.

[44] Mechthild Stoer. *Design of survivable networks.* Lecture Notes in Mathematics. Springer, Berlin, 1992.

[45] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.

[46] M. M. Syslo. On cycle bases of a graph. *Networks*, 9(2):123–132, 1979.

[47] Arie Tamir. On totally unimodular matrices. *Networks*, 6(4):373–382, 1976.

[48] Toshimasa Watanabe and Akira Nakamura. Edge-connectivity augmentation problems. *J. Comput. Syst. Sci.*, 35(1):96–144, August 1987.

[49] Shi Xiao and Gaoxi Xiao. On imperfect node protection in complex communication networks. *Journal of Physics A: Mathematical and Theoretical*, 44(5):055101, 2011.

[50] XO Communications. Network map. Available at `http://www.xo.com/about/network/Pages/maps.aspx`.