Enhancing Network Robustness via Shielding

Jianan Zhang, *Student Member, IEEE*, Eytan Modiano, *Fellow, IEEE*, and David Hay, *Member, IEEE*

***Abstract*— We consider shielding critical links to enhance the robustness of a network, in which shielded links are resilient to failures. We first study the problem of increasing network connectivity by shielding links that belong to small cuts of a network, which improves the network reliability under random link failures. We then focus on the problem of shielding links to guarantee network connectivity 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 pair under a general failure model, and exploit geometric 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. We also apply simulated annealing to obtain near-optimal solutions in much shorter time. Finally, we extend the algorithms to guarantee partial network connectivity, and observe significant reduction in the shielding cost, especially when the geographical failure region is small.**

***Index Terms*— Connectivity, network robustness, geographical failure, shielding.**

I. Introduction

COMMUNICATION networks are subject to natural disasters and attacks, such as hurricanes, earthquakes, and electromagnetic pulse attacks [1]. Network failures may result in tremendous financial loss and hinder effective recovery to the affected regions. Therefore, it is important to design robust networks that can withstand disasters or attacks.

Several metrics measure the performance of a network. The most basic metric is connectivity, without which it is impossible to support any application that requires data communication through the network. In case of network failures, one cannot expect the network to support the same amount of traffic as before the failure. Low priority applications can be throttled to give higher priority to critical applications. Thus, in this paper, we focus on guaranteeing network connectivity after failures, and assume that the network is able to use limited resources to support critical applications through service differentiation.

Manuscript received April 3, 2016; revised October 17, 2016; accepted February 28, 2017; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Eryilmaz. Date of publication April 24, 2017; date of current version August 16, 2017. This work was supported in part by the NSF under Grant CNS-1017800, in part by the DTRA under Grant HDTRA-09-1-0050 and Grant HDTRA1-14-1-0058, in part by the U.S.-Israel Binational Science Foundation, and in part by the Israeli Centers of Research Excellence Program under Grant 4/11. *(Corresponding Author: Jianan Zhang.)*

J. Zhang and E. Modiano are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jianan@mit.edu; modiano@mit.edu).

D. Hay is with the Rachel and Selim Benin School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel (e-mail: dhay@cs.huji.ac.il).

Digital Object Identifier 10.1109/TNET.2017.2689019

Previous research considers geographical failures [2]–[5] and general failures [6]–[8], in addition to independent random failures [9], [10], to assess the network robustness. Geographical failure models capture the effects of natural disasters and physical attacks; e.g., all links in a geographical failure region are destroyed. Under the general failure model, an arbitrary set of links may fail; e.g., each failure may affect a specified set of links, whose number and locations are determined by the nature of the failure.

A common approach to design robust networks is through redundancy and backup routes. (See [11], [12] for a survey of protection techniques for optical networks.) Graph augmentation algorithms, which aim to increase the connectivity of a graph by adding new edges, can be used for adding redundancy. Exact polynomial-time algorithms have been proposed to solve the augmentation problem if new edges can be added between any pair of nodes and have the same cost [13], [14]. Approximation algorithms have been developed to solve the weighted version of the augmentation problem [15], [16]. Mathematical programming techniques are often used to develop stronger formulations by adding cutting planes to solve the general augmentation problem [8].

An alternative approach to design robust networks, which we consider in this paper, is through shielding critical links. Shielded network infrastructure can survive disasters and attacks. Previous research suggested strengthening cables to resist physical attacks [12], and upgrading or covering vulnerable components to resist electromagnetic pulse attacks [17]. More robust optical fibers and cables are being developed to improve network reliability [18], [19], and recently, Google plans to reinforce undersea cables to resist shark attacks [20].

A. Related Work

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 enhance the network robustness. The idea of shielding has been applied to solve problems arising from various applications [21]–[27]. Liu *et al.* [21] studied a road network retrofit problem, and used a two-stage stochastic programming approach to decide which roads to retrofit to minimize the average performance loss incurred by a disaster. Snyder *et al.* [22], Church *et al.* [23], and Brown *et al.* [24] studied problems of fortifying facilities to minimize the transportation cost in supply chains and the failures of critical infrastructure. Dziubiński and Goyal [25] designed optimal network topologies to maximize the utility less costs, given shielding cost, link construction cost, and utility of network connectivity, under the assumption of uniform costs for all links.

Cunningham [26] studied the reinforcement problem in order to maximize the cost of destroying a network by an adversary. Alashaikh *et al.* [27] studied the problem of improving network reliability by increasing the reliability of edges in a spanning tree. A preliminary version of this paper appeared in [28].

The shielding problem that we study is closely related to the graph augmentation problem where the new edges to be added must be *parallel* to the existing edges. Unlike the polynomially solvable graph augmentation problem where edges can be added between any pair of nodes, determining the minimum number of parallel edges to be added to increase the edge-connectivity of a graph is NP-hard [29]. Shielding an edge can be viewed as adding a sufficiently large number of parallel edges at a fixed cost, so that a shielded edge cannot be contained in any cut of a moderate size. To this end, the shielding problem shares similarities with the fixed-charge problems [30], [31], where there is a fixed cost of using an edge regardless of the amount of flow carried on the edge.

Our study is closely related to the work in [32], in which the authors proved the NP-completeness, and developed an exact algorithm to obtain the minimum cost reinforcement (i.e., shielding) that guarantees one or multiple source-destination (SD) pairs are connected after fewer than k removals of unreinforced edges. Based on a directed graph structure that represents all the minimum SD cuts, the algorithm in [32] selects a set of edges such that every cut whose size is smaller than k contains at least one selected edge. The algorithm is doubly exponential in k. In contrast, we consider the connection between cuts and network flows. By interpreting "shielding" as increasing the edge capacity, we formulate a mixed integer linear program (MILP) which is simpler, contains a small number of variables and constraints, and can be solved within a short time for realistic networks. For the special case of increasing the edge-connectivity of an SD pair by one, we develop a polynomial algorithm which runs faster than the algorithm in [32]. Moreover, we consider shielding links to resist geographical and general failures.

B. Our Contributions

Our objective is to design robust communication networks by shielding critical links of the network. We develop algorithms to obtain the optimal shielding to resist both independent random failures and correlated failures. Throughout this paper we consider link (edge) failures and assume that all nodes are resilient to failures. Our main contributions include:

1) Optimal Shielding Under Independent Random Failures: The robustness of a network under independent random edge failures can be measured by its *edge-connectivity*, the minimum number of edge removals that separate nodes into two disjoint sets. By shielding critical edges and restricting the removed edges to be from the unshielded set, the network can resist more edge removals and has larger edge-connectivity. In Section II, we study an equivalent minimum cost fixed-charge network flow problem, where shielded edges have larger capacities, and develop a MILP to obtain the optimal shielding that increases the edge-connectivity of a pre-selected SD pair. Then by a simple extension, we obtain the optimal shielding to increase the edge-connectivity of a network.

2) Optimal Shielding Under Correlated Failures: While increasing the edge-connectivity is the key to improving network robustness under random failures, real world disasters and attacks often result in correlated failures. We consider both geographical failures and general failures.

In Section III, we aim to shield links using the minimum cost to guarantee the connectivity of an SD pair and a network after any single failure event. To guarantee the connectivity of an SD pair, we develop a MILP for the general shielding problem, and identify the property of the optimal shielding in the case of geographical failures, which allows us to decompose the shielding problem to multiple subproblems. We then extend the MILP to obtain the optimal shielding that guarantees network connectivity. By identifying the separable structure of the MILP, we are able to apply the Benders decomposition technique [33] to reduce the running time and solve network shielding problems of realistic sizes. A heuristic based on simulated annealing further reduces the running time significantly while outputting a feasible solution with a slightly higher shielding cost.

Finally, in Section IV, we relax the connectivity requirement to allow for partial connectivity of a network. We observe that the shielding cost can be significantly reduced if the connectivity requirement is slightly relaxed.

II. Increasing the Edge-Connectivity

A communication network can be represented by a graph, where nodes represent routers and edges represent optical fibers. In a communication network, a shielded fiber is resilient to attacks. To model the resilience, in a graph, a "shielded" edge cannot be removed. Shielding an edge in a graph can be interpreted in different contexts. For example, for the purpose of network connectivity, a shielded edge can be viewed as having been contracted, so that the two end nodes of a shielded edge are merged and are always connected. In this section, a shielded edge is viewed as having a sufficiently large capacity, so that small cuts do not contain shielded edges.

If all the edges in a graph have unit capacity, the edge-connectivity of the graph equals the *value of the minimum cut*, which is the minimum total capacity of edges whose removals separate nodes into two disjoint sets. If a shielded edge has a sufficiently large capacity, all the minimum cuts in a shielded graph only contain unshielded unit-capacity edges. Thus, the value of the minimum cut in a shielded graph is exactly the edge-connectivity of the shielded graph where *only unshielded edges can be removed*. The following lemma interprets shielding an edge as increasing the edge capacity, and is the basis of the analysis in this section.

Lemma 1: Consider a set of shielded edges $E^* \subseteq E$ that increase the edge-connectivity of an undirected graph $G(V, E)$ from k to k'. If instead of shielding, the capacity of each edge in E^* was increased from 1 to $k' - k + 1$, and the capacity of each edge in $E \setminus E^*$ was 1, then the value of the minimum cut of $G(V, E)$ would be k'.

Proof: Since shielding the edges E^* yields an edge-connectivity of k', the minimum subset of $E \setminus E^*$ that disconnects G has size k' (comprising only of unshielded edges).

We next show that all the other cuts in G have value at least k'. Consider any cut C in G and let i denote the number of shielded edges in C. On the one hand, $|C|$ is at least k because this is the edge-connectivity of G considering all edges. On the other hand, i is at least one because C must have at least one shielded edge. Thus, the value of C is $(|C| - i) + i(k' - k + 1) \geq (k - i) + i(k' - k + 1) = k' + (i - 1)(k' - k) \geq k'$. ■

A. Increasing the Edge-Connectivity of an SD Pair

The edge-connectivity of a specific pair of nodes (namely, an SD pair) in a shielded graph is the minimum number of unshielded edge removals that disconnect the SD pair. We formally state the optimal shielding problem that increases the edge-connectivity of an SD pair as follows.

Minimum Cost Shielding to Increase the Edge-Connectivity of an SD Pair: Given an undirected graph $G(V, E)$, an SD pair $(s, d \in V)$ with edge-connectivity k, a cost c_{ij} for each edge (i, j), and an objective edge-connectivity $k' > k$, we aim to shield a set of edges with the minimum cost, such that the SD pair is connected after removing any $k' - 1$ unshielded edges (i.e., the edge-connectivity of the SD pair is increased to k').

Based on the same proof as Lemma 1 and considering cuts that separate an SD pair, it is easy to see that by increasing the capacity of a shielded edge from 1 to $k' - k + 1$, the value of the minimum SD cut would be increased from k to k', as long as the shielding increases the edge-connectivity of an SD pair from k to k'. On the other hand, if the shielding merely increases the edge-connectivity of the SD pair to a value smaller than k', removing fewer than k' unshielded edges disconnects the SD pair, in which case there exists an SD cut of value smaller than k'. Thus, the optimal shielding problem is equivalent to the problem of increasing the value of the minimum SD cut to k', by increasing some edge capacities (from 1 to $k' - k + 1$) using the minimum cost. A classical way to tackle such a cut problem is by working on its corresponding network flow problem, which is, in our case, a (hard) variation of *minimum cost flow* problem (also known as the fixed-charge problem [30], [31]), defined as follows:

Minimum Cost Shielding Restated as a Flow Problem: Given an undirected graph $G(V, E)$, an SD pair $(s, d \in V)$ with edge-connectivity k, and an objective $k' > k$, we aim to find the minimum cost *flow* of value k' between the SD pair, where the cost for an edge (i, j) to carry up to one unit flow is zero, and the cost for an edge (i, j) to carry more than one unit flow, up to $k' - k + 1$ units flow, is c_{ij}.

It is important to notice that any edge that carries more than one unit of flow must be shielded, as originally the capacity of an edge is 1. Intuitively, in order to save cost, the solution will try to avoid using more than one unit of flow in any edge and not incur its cost c_{ij}.

We note that our problem is significantly different from the classical minimum cost flow problem, as the definition of the cost (of an edge) is not linear in the flow that it carries (we use a step function instead). Such problems are known as fixed-charge problems, which are NP-hard in general [34]. Moreover, the NP-hardness of the minimum cost shielding problem has also been proved in [32], by a reduction from the Steiner tree problem.

We obtain the exact solution of the minimum cost shielding problem using MILP. Let $\{(i, j) | h_{ij} = 1\}$ be the set of shielded edges. Let x_{ij} be the amount of flow carried on edge (i, j), which is at most 1 on an unshielded edge and may take a sufficiently large value up to $M + 1$ on a shielded edge. We set $M = k' - k$ based on Lemma 1. Note that larger M still gives a valid formulation. The importance of Lemma 1 is on improving the strength of the formulation. The formulation is stronger (i.e., the integrality gap is smaller) for smaller M. Thus, using a small M reduces the solving time of the MILP.

$$\min \sum_{(i,j) \in E} c_{ij} h_{ij} \tag{1}$$

$$\text{s.t.} \sum_{\{j|(i,j)\in E\}} x_{ij} - \sum_{\{j|(j,i)\in E\}} x_{ji} = \begin{cases} k', & \text{if } i = s \\ -k', & \text{if } i = d \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$x_{ij} \leq 1 + M h_{ij} \quad \forall (i, j) \in E \tag{3}$$

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

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

Flow conservation constraints (2) guarantee k' units flow from the source to the destination. Each edge (i, j) is undirected and it is shielded if either $h_{ij} = 1$ or $h_{ji} = 1$. However, constraints (3) only allow the edge to carry more than one unit flow along $i \rightarrow j$ direction if $h_{ij} = 1$. This formulation is valid, because h_{ij} and h_{ji} cannot both be equal to one in the optimal solution. 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, given that only a single commodity of flow is routed. Therefore, either h_{ij} or h_{ji} is nonzero to guarantee that the link (i, j) is shielded and may carry up to $M + 1$ units flow. The optimal shielding cost is given by $\sum_{(i,j)\in E} c_{ij} h_{ij}$.

The formulation contains only $|V| + |E|$ constraints in addition to the boundary constraints. The size of this formulation is much smaller compared with the cut formulation (i.e., every cut of size smaller than k' must contain at least one shielded edge) for solving the shielding problem, where there are an exponential number of cut constraints.

It is interesting to note that MILP (1) is polynomially solvable when $M = 1$. Thus, the minimum cost shielding to increase the edge-connectivity of an SD pair by one can be obtained in polynomial time.

Theorem 1: MILP (1) is polynomially solvable when $M = 1$, and the optimal solution is given by its linear programming relaxation.

Proof: It suffices to show that the linear programming (LP) relaxation of MILP (1) has an integral optimal solution when $M = 1$. A sufficient condition 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

$$|\sum_{i\in I_1} A_{ij} - \sum_{i\in I_2} A_{ij}| \leq 1, \quad \forall j \in \{1, 2, \ldots, n\}. \tag{4}$$

Each variable x_{ij} appears in two constraints in (2) with coefficients 1 and -1. Let A_1 denote the coefficient matrix of constraints (2). For any subset I^a of rows of A_1, $I_1 = I^a$, $I_2 = \emptyset$ clearly satisfies condition (4). Among the constraints (3), each variable x_{ij} and h_{ij} only appear in one constraint. Let A_2 denote the coefficient matrix of constraints (3) 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 the row in I^b that has nonzero coefficient of x_{ij} can be either in I_1 or I_2, without violating the constraint 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 (4) is satisfied and the constraint coefficient matrix is TU. ■

Indeed, there is an efficient combinatorial algorithm to compute the optimal shielding to increase the edge-connectivity of an SD pair in graph G by one. Recall the minimum cost flow interpretation for the shielding problem. To obtain the minimum cost fixed-charge flow on G, we construct another graph G' as follows. Graph G' has the same nodes as G, and has an edge $(i, j)^o$ if and only if there is an edge (i, j) between the same pair of nodes in G. Edges $(i, j)^o$ in G' have unit capacity and zero cost. In addition, a parallel edge $(i, j)^p$ is added adjacent to each zero-cost edge $(i, j)^o$ in G', and has unit capacity and cost c_{ij} *per unit flow*.

We prove that the optimal shielding can be obtained by applying a classical minimum cost k'-flow algorithm to G'.

Theorem 2: The minimum cost shielding to increase the edge-connectivity of an SD pair from k to $k' = k + 1$ can be computed in $O(k'|E| + |V| \log |V|)$ time, where $|E|$ is the number of edges and $|V|$ is the number of nodes in the graph.

Proof: Consider an *integer-valued* (classical) minimum cost k'-flow in G' from the source to the destination, where each edge carries either zero or one unit flow. An integer-valued minimum cost k'-flow exists, given that k' is an integer and all the edges have integer capacities [33].

An integer-valued minimum cost k'-flow in G' corresponds to a k'-flow in G with the same cost and along the same route. In G', the total cost of the flow equals the cost of using the parallel edges $(i, j)^p$, given that $(i, j)^o$ are free. If an edge $(i, j)^p$ carries unit flow, then $(i, j)^o$ must also carry unit flow, because otherwise re-routing the flow from $(i, j)^p$ to $(i, j)^o$ yields a flow that has smaller cost. Therefore, each parallel

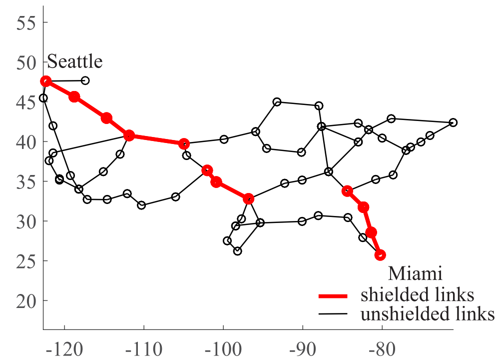


Fig. 1. Optimal shielding to increase the edge-connectivity of an SD pair Seattle-Miami from 2 to 3.

edge $(i, j)^p$ that carries unit flow in G' corresponds to an edge (i, j) in G that carries two units flow, and incurs a cost c_{ij} equal to the cost of (i, j) that carries two units flow in G (i.e., the shielding cost of (i, j)).

Moreover, it is clear that any flow in G maps to a flow in G' with no larger cost along the same route. In G, any edge (i, j) that carries more than one and up to two units flow incurs cost c_{ij}. In G', the cost for $(i, j)^o$ to carry unit flow is zero, and the cost for $(i, j)^p$ to carry up to one unit flow is at most c_{ij}.

Thus, the minimum cost fixed-charge k'-flow in G has the same cost and route as the integer-valued minimum cost k'-flow in G'. The classical minimum cost k'-flow algorithm on G' solves the minimum cost fixed-charge k'-flow problem in G, which yields the optimal shielding that increase the edge-connectivity of the SD pair by one.

The minimum cost k'-flow on G' can be computed by finding k' shortest augmenting paths, where the length of an edge equals its cost, given that each edge in G' has unit capacity [35]. Moreover, there exists k units flow of zero cost in G', because the max-flow between the SD pair is k by only using zero-cost edges $(i, j)^o$. The k zero-length paths can be computed by k augmenting path algorithms in $O(k|E|)$ time. The computation of the last shortest augmenting path takes $O(|E| + |V| \log |V|)$ time by Dijkstra's algorithm [36]. Therefore, the total running time is $O(k'|E| + |V| \log |V|)$. ■

This algorithm improves upon the algorithm in [32] which runs in $O(k'^3|E| + |V| \log |V|)$ (or $O(k'^3|E|)+$ shortest path) time to increase the SD edge-connectivity by one. The improvement is significant when the edge-connectivity is large.

To demonstrate the application of our approach, we solve the MILP to obtain the optimal shielding to increase the edge-connectivity of an SD pair Seattle-Miami in the XO communication backbone network which consists of 60 nodes and 71 edges [37]. The cost of shielding each edge is represented by the length of the edge (in longitude/latitude degree unit labeled on the axes). The edge-connectivity of the SD pair is $k = 2$ before shielding. To increase the edge-connectivity by one ($k' = 3$), both MILP (1) and the algorithm of Theorem 2 output a solution of cost 35.2, depicted in Fig. 1. The shielding cost increases as k' increases: 47.8 for $k' = 4$ and 51.5 for $k' = 5$. The shielded edges are depicted in Figs. 2 and 3, respectively. Interestingly, for $k' = 5$, the shielded edges form the shortest path between the SD pair. Thus, it guarantees that removing any number of unshielded edges would not

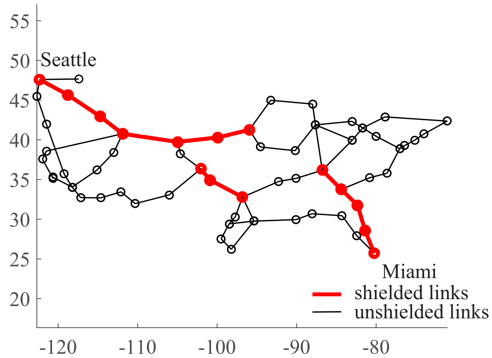


Fig. 2. Optimal shielding to increase the edge-connectivity of an SD pair Seattle-Miami from 2 to 4.

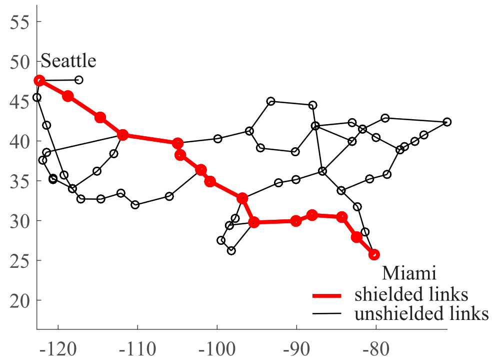


Fig. 3. Optimal shielding to increase the edge-connectivity of an SD pair Seattle-Miami from 2 to 5.

disconnect the SD pair. The running times to solve MILP (1) are less than 1 second to obtain the above results (executed on a desktop PC with Intel® Xeon® 2.67 GHz CPU and 4 GB RAM). The solutions take a short time because only a single commodity flow needs to be carried through the network between a single SD pair, and the number of constraints is small.

B. Increasing the Edge-Connectivity of a Network

We consider the optimal shielding to increase the edge-connectivity of a network, formally stated as follows.

Minimum Cost Shielding to Increase the Edge-Connectivity of a Network: Given an undirected graph $G(V, E)$ with edge-connectivity k, a cost c_{ij} for each edge (i, j), and an objective edge-connectivity $k' > k$, we aim to shield a set of edges with the minimum cost, such that $G(V, E)$ is connected after removing any $k'-1$ unshielded edges (i.e., the edge-connectivity of $G(V, E)$ is increased to k').

Unlike the polynomially solvable optimal shielding problem that increases the edge-connectivity of an SD pair by one, the optimal shielding problem that increases the edge-connectivity of a graph by one is NP-hard. Consider the connection between the shielding problem and the graph augmentation problem. If the objective of shielding is to increase the edge-connectivity from k to $k+1$, shielding an edge can be viewed as augmenting *one* parallel edge (or increasing the edge capacity by one in the network flow problem). The problem of determining the minimum number of parallel edges to be added to a graph in order to increase the edge-connectivity of a graph by one has been shown to be NP-hard [29]. Therefore, the minimum cost shielding problem that increases the edge-connectivity of a graph by one is NP-hard.

1) Upper Bound on the Optimal Shielding Cost: We provide an upper bound on the optimal shielding cost, by developing a shielding algorithm that achieves the desired edge-connectivity using a potentially larger shielding cost. We first prove that shielding some part of a spanning tree of the original graph is sufficient to increase its edge-connectivity.

Theorem 3: Given any spanning tree $T(V, E_T \subseteq E)$, shielding the edges in E_T that belong to cuts of $G(V, E)$ that have size smaller than k' is sufficient to increase the edge-connectivity of $G(V, E)$ to k'.

Proof: To prove that G is connected after removing fewer than k' unshielded edges, it is equivalent to prove that each cut of size smaller than k' contains at least one shielded edge.

Let E_C denote the union of edges in cuts of size smaller than k'. Let $E_{NC} = E \setminus E_C$ denote the other edges. Let E_T denote the edges in a spanning tree T. Let $E_S = E_T \cap E_C$ denote the shielded edges. $E_T = E_T \cap E = E_T \cap (E_C \cup E_{NC}) = (E_T \cap E_C) \cup (E_T \cap E_{NC}) \subseteq (E_T \cap E_C) \cup E_{NC} = E_S \cup E_{NC}$. Namely, the union of E_S and E_{NC} is a superset of E_T, and $G(V, E_S \cup E_{NC})$ is connected.

Suppose that there exists a cut C_0 that contains fewer than k' unshielded edges and no shielded edge. By definition, $C_0 \subseteq E_C$, and $G(V, E \setminus C_0)$ is disconnected. Since C_0 do not contain any shielded edges, $E_S \subseteq E_C \setminus C_0$. Thus, $E_S \cup E_{NC} \subseteq (E_C \setminus C_0) \cup E_{NC} = E \setminus C_0$. The fact that $G(V, E_S \cup E_{NC})$ is connected contradicts with that $G(V, E \setminus C_0)$ is disconnected.

To conclude, shielding E_S guarantees that G is connected after removing fewer than k' unshielded edges, and increases the edge-connectivity of G to k'. ■

The following algorithm computes a tree T in which the total shielding costs of the edges that belong to cuts of size smaller than k' is minimized. The algorithm increases the edge-connectivity of a graph to at least k'. We call the obtained shielding cost the *spanning tree upper bound*.

Algorithm 1 Spanning Tree Shielding Algorithm

1) Identify edges $E_C(k')$ that belong to cuts of $G(V, E)$ that have size smaller than k'.
2) Assign each edge in $E_C(k')$ a cost equal to its shielding cost. Assign all the other edges zero cost.
3) Compute the minimum cost spanning tree, and shield its edges that have positive costs.

Step 1 can be implemented efficiently if $\beta = k'/k$ is not very large, where k is the edge-connectivity of $G(V, E)$. With high probability, all the minimum cuts can be identified in $O(|V|^2 \log^3 |V|)$ time, and all the cuts that have value smaller than k' can be identified in $O(|V|^{2\beta} \log^2 |V|)$ time [38]. The minimum spanning tree can be computed in $O(|E| + |V| \log |V|)$ time (e.g., Prim's algorithm).

Remark: The spanning tree upper bound is tight if the edge-connectivity of G is increased from 1 to 2 ($k = 1, k' = 2$). In the Spanning Tree Shielding Algorithm, when costs are

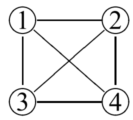


Fig. 4. Shielding the two diagonals increases the network edge-connectivity by one.

assigned to edges, an edge has positive cost if its removal disconnects G (i.e., if it is a cut edge), and has zero cost otherwise. All the spanning trees in G each include all the cut edges while only differ in the zero-cost edges, and they have the same cost. Shielding all the positive-cost edges in any spanning tree is optimal, because the failure to shield any cut edge would not increase the edge-connectivity of G by one.

However, the spanning tree upper bound is not necessarily tight in general, as shown by the following example. Consider a clique of four nodes as in Fig. 4. The edge-connectivity is three, and all the edges belong to minimum cuts of size three. Instead of shielding a spanning tree, only two diagonal edges (or any maximum matching) need to be shielded in order to increase the graph edge-connectivity by one. For the special case of increasing the edge-connectivity by one, algorithms in [15] and [16] can be used to compute a feasible shielding with at most twice the optimal cost.

2) Exact Solution of the Minimum Cost Shielding Problem Using MILP: The minimum cost shielding problem of increasing the edge-connectivity of a network can be equivalently stated as a flow problem, analogous to the minimum cost shielding problem of increasing the edge-connectivity of an SD pair. We extend MILP (1) to formulate the optimal shielding problem that increases the edge-connectivity of a network. One difference from MILP (1) is that the flows between *all* pairs of nodes should each be at least k', guaranteed by constraints (6). In fact, it suffices to check the connectivity between one fixed node and every other node, a total of $|V|-1$ pairs. Another difference is that x_{ij}^{sd} and $x_{ji}^{s'd'}$ on two directions of an edge may be both greater than one for different SD pairs. Constraints (7) enforce that $h_{ij}=h_{ji}=1$ for any shielded edge (i,j), and the double counting is fixed by reducing the objective value by half.

$$\min \sum_{(i,j)\in E} c_{ij}h_{ij}/2 \tag{5}$$

$$\text{s.t.} \sum_{\{j|(i,j)\in E\}} x_{ij}^{sd} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{sd} = \begin{cases} k', & \text{if } i=s \\ -k', & \text{if } i=d \\ 0, & \text{otherwise} \end{cases} \quad \forall s,d \tag{6}$$

$$x_{ij}^{sd} \le 1 + Mh_{ij} \quad \forall (i,j)\in E, s,d$$
$$h_{ij}-h_{ji}=0 \quad \forall (i,j)\in E \tag{7}$$
$$x_{ij}^{sd} \ge 0 \quad \forall (i,j)\in E, s,d$$
$$h_{ij}=\{0,1\} \quad \forall (i,j)\in E$$

We obtain the optimal shielding to increase the edge-connectivity of the XO network by solving MILP (5). Table I shows that the running time increases significantly as the objective edge-connectivity increases. This is consistent with

TABLE I

INCREASING THE EDGE-CONNECTIVITY OF THE XO NETWORK

Objective edge-connectivity k'	2	3	4	5
Optimal cost by MILP	4.9	134.1	147.7	163.4
Spanning tree upper bound	4.9	153.7	173.4	173.4
Time of solving MILP (s)	0.13	10.29	33.73	1143.68

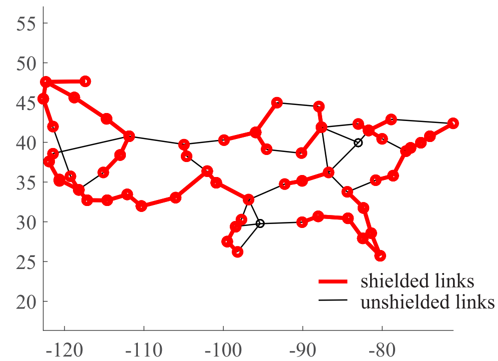


Fig. 5. Optimal shielding to increase the edge-connectivity of the XO network from 1 to 3.

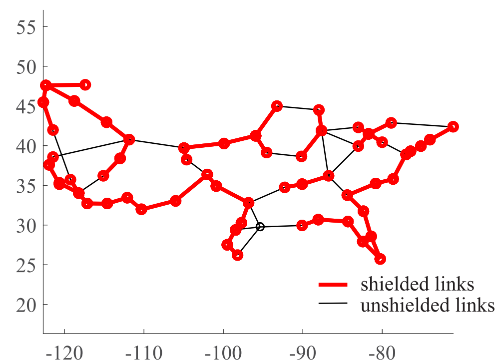


Fig. 6. Optimal shielding to increase the edge-connectivity of the XO network from 1 to 4.

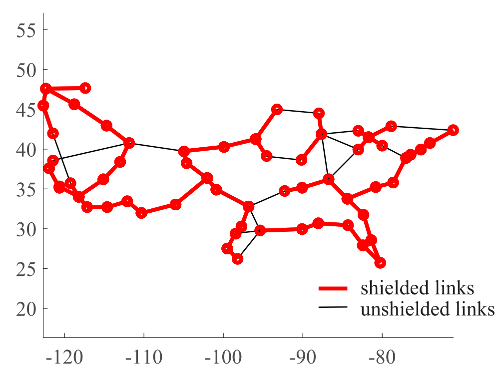


Fig. 7. Optimal shielding to increase the edge-connectivity of the XO network from 1 to 5.

the fact that the formulation is stronger for smaller values of $M=k'-k$, where k' is the objective edge-connectivity and k is the original edge-connectivity ($k=1$ in this example). The optimal shielding for $k'=3$ is depicted in Fig. 5. From the figure, we observe that many edges need to be shielded if the original edge-connectivity between most adjacent SD pairs are smaller than the objective edge-connectivity. The optimal shielding for $k'=4$ and 5 are depicted in Figs. 6 and 7, respectively. In this example, most shielded edges for a smaller edge-connectivity (in Fig. 5) are still shielded in order to achieve a larger edge-connectivity (in Figs. 6 and 7).

Moreover, we compare the optimal cost with the spanning tree upper bound obtained by Algorithm 1. To increase the edge-connectivity of the XO network by one, only one edge needs to be shielded at cost 4.9, which is identical to the optimal shielding obtained by the MILP. To increase the edge-connectivity by two, we enumerate all minimal cuts of size one and two and shield the cut edges that belong to a minimum spanning tree where the cost of an edge not in these cuts is zero. The shielding cost is 153.7, which is 15% higher than the optimal cost obtained by the MILP. To increase the edge-connectivity by three, Algorithm 1 shields the entire minimum spanning tree, which has cost 173.4 and is 17% higher than the optimal cost. To further increase the edge-connectivity, the upper bound stays fixed while the optimal shielding cost keeps increasing until reaching the bound. From the results we observe that the bound is tighter if there are not many edges in the small cuts when the objective edge-connectivity k' is small, or if k' is large and almost an entire spanning tree needs to be shielded. Indeed, if k' is larger than the size of the maximum cut of the graph, the spanning tree upper bound is the same as the optimal solution.

III. Guaranteeing the Connectivity Under Correlated Failures

In the previous section, we considered the minimum cost shielding to increase the edge-connectivity, which makes the network more robust under independent random link failures. In the rest of this paper, we consider the minimum cost shielding to resist a single geographical or general failure that can represent a real-world disaster or attack.

A geographical failure is characterized by a disk region with a given radius, and may occur anywhere in the network. All the unshielded links that are intersected or covered by the disk region are removed. A general failure is characterized by a set of links that the failure affects. The possible failures are described explicitly. We study the optimal shielding to guarantee that an SD pair and a network are connected after any single geographical and general failure.

A. Guaranteeing the Connectivity of an SD Pair

It suffices to shield links to guarantee that a path between an SD pair will exist after any failure event. Clearly, if one link can fail at a time, a link needs to be shielded if and only if its removal disconnects the SD pair. In contrast, if multiple links fail at the same time and their failures disconnect the SD pair, not all the failed links need to be shielded. We aim to compute the minimum cost shielding to guarantee the connectivity of an SD pair after any failure event under the general and geographical failure models in this section.

1) Shielding Under the General Failure Model: Under the general failure model, a failure event is specified by a set of affected links. In $G(V,E)$, each failure event $z \in Z$ affects links $E^{(z)}$, where Z is a set of possible failure events. The objective is to shield a set of links E^* using the minimum shielding cost, to guarantee that an SD pair

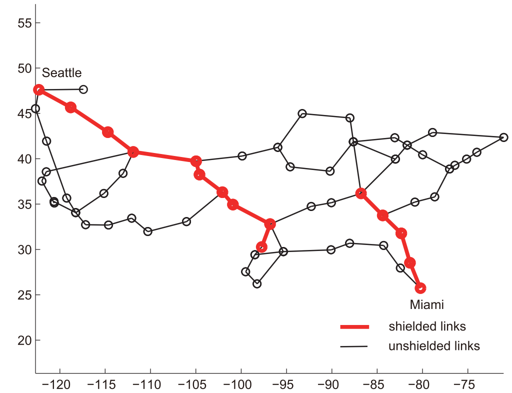


Fig. 8. Optimal shielding for an SD pair under the failure model where all the links adjacent to any two nodes are affected by a failure.

(s,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 formulated by a MILP. Let $t_{ij}^{(z)}$ indicate whether link (i,j) is affected by failure z. The set of links affected by failure z is $E^{(z)} = \{(i,j)|t_{ij}^{(z)} = 1\}$. The decision variables are $x_{ij}^{(z)}$ and h_{ij}, which represent the amount of flow carried by link (i,j) after failure z and whether link (i,j) is shielded, respectively. The set of shielded links is $E^* = \{(i,j)|h_{ij}=1\}$. The minimum shielding cost to resist any possible failure $z \in Z$ is given by the following MILP.

$$\min \sum_{(i,j)\in E} c_{ij}h_{ij} \tag{8}$$

$$\text{s.t.} \sum_{\{j|(i,j)\in E\}} x_{ij}^{(z)} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(z)} = \begin{cases} 1, & \text{if } i = s \\ -1, & \text{if } i = d \\ 0, & \text{otherwise} \end{cases} \quad \forall z \in Z \tag{9}$$

$$x_{ij}^{(z)} - h_{ij} \le 1 - t_{ij}^{(z)} \quad \forall (i,j) \in E, z \in Z \tag{10}$$

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

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

Since we consider a connectivity problem, only one unit flow need to be carried from s to d after a failure, which is guaranteed by the flow constraints (9). Constraints (10) guarantee that if failure z occurs and affects link (i,j) ($t_{ij}^{(z)}=1$), one 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 one unit flow if failure z occurs, regardless of whether it is shielded.

As an example, we consider a failure model where links adjacent to any pair of nodes are affected by a failure event, and study the optimal shielding to guarantee the connectivity of Seattle-Miami in the XO network. The number of failures is $\binom{60}{2}$ (i.e., this model allows for link failures adjacent to 2 nodes, and the total number of nodes is 60). The optimal shielding is represented by the thick links with total cost 46.0 in Fig. 8.

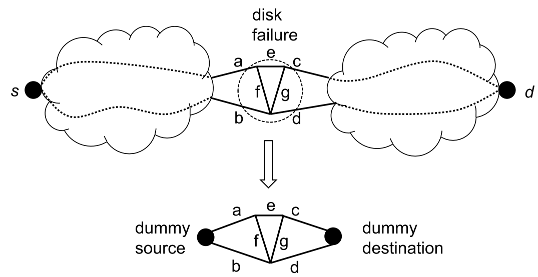


Fig. 9. A single bottleneck.

2) Shielding Under the Geographical Failure Model: We aim to shield a set of links such that an SD pair (s, d) are connected after any disk failure of a given radius. It is necessary to identify the set of geographical failure regions where a disk failure disconnects the SD pair. We call such regions *bottleneck* regions. Although the number of possible failure regions is large, multiple failures can be represented by one failure that dominates them, where a failure dominates another if it affects all the links that are 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 [2]–[4]. In the following, we exploit properties of bottleneck regions to decompose the shielding problem into several subproblems, each of which consists of one or more bottleneck regions and can be solved independently.

First, consider a single bottleneck region. We use Algorithm 2 to compute a shielded path that guarantees the connectivity of an SD pair after a failure occurs at the bottleneck region, and illustrate the algorithm by the example in Fig. 9. In Step 1 of the algorithm, the end nodes that are merged to a dummy source can be found by first removing the links intersected or covered by the bottleneck region and then performing a depth-first-search from the source. Thus, Step 1 takes $O(|E|)$ time, where $|E|$ is the number of links in the network. Step 2 of the algorithm also takes $O(|E|)$ time, by performing a depth-first-search from the destination. The running time of Step 3 depends on the objective path. For example, the shortest path can be computed in $O(|E| + |V| \log |V|)$ time, where $|V|$ is the number of nodes.

Algorithm 2 Bottleneck Shielding Algorithm

Let E_B denote the links intersected or covered by the bottleneck region. Let V_B denote the end nodes of E_B.

1) Find the end nodes in V_B that are connected to the source without going through any link in E_B (end nodes of links a, b in Fig. 9). Merge the end nodes to form a dummy source.
2) Find the end nodes in V_B that are connected to the destination without going through any link in E_B (end nodes of links c, d in Fig. 9). Merge the end nodes to form a dummy destination.
3) Compute a path P between the dummy SD pair. Shield the links in $P \cap E_B$.

The following fact proves the correctness of Algorithm 2.

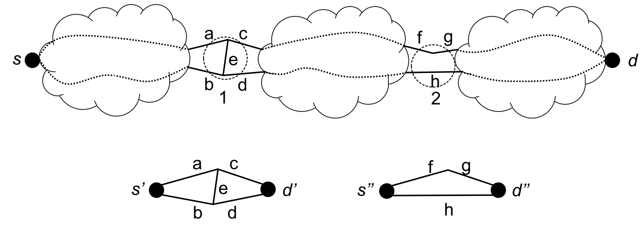


Fig. 10. Non-overlapping bottlenecks.

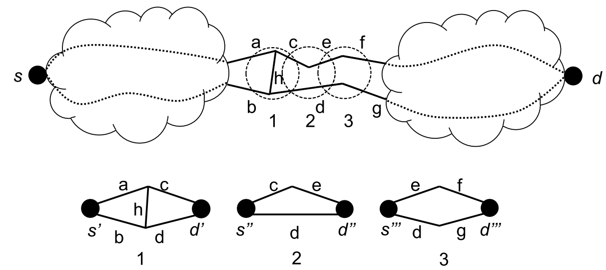


Fig. 11. Overlapping bottlenecks.

Proposition 1: A shielded path between the dummy SD pair (defined in Algorithm 2) is necessary and sufficient to guarantee that the original SD pair is connected after a geographical failure that occurs at the bottleneck region.

Proof: Sufficiency: Consider a geographical failure that occurs at the bottleneck region and affects links E_B. Let P be a path between the dummy SD pair. Shielding links in $P \cap E_B$ guarantees that P survives. Moreover, the dummy source (destination) is connected to the original source (destination) without using E_B. Therefore, the original SD pair is connected.

Necessity: Consider an SD path P_0 that survives a failure at the bottleneck region. Clearly $P_0 \cap E_B$ are shielded. Let $\{l_1, \ldots, l_p\}$ be an ordered sequence of links in P_0. Let l_i be the first link in P_0 that belongs to E_B. One end node v of l_i is connected to the source without going through E_B. Therefore, v is merged to the dummy source. Let l_j be the last link in P_0 that belongs to E_B. One end node of l_j is merged to the dummy destination. The sub-path $P = \{l_i, \ldots, l_j\}$ connects the dummy SD pair, and $P \cap E_B \subseteq P_0 \cap E_B$ are shielded. ■

If there is only one bottleneck between an SD pair, the optimal shielding is to shield the shortest path between the dummy SD pair, where the length of each link equals its shielding cost. In general, there may be multiple bottlenecks. In order to guarantee the connectivity of the SD pair after any disk failure, one needs to shield a path across every bottleneck. If the bottlenecks are disjoint and do not share common links (Fig. 10), shielding the shortest path between each dummy SD pair is optimal.

However, different bottlenecks may overlap and intersect common links (Fig. 11). Shielding links for one bottleneck may affect the shielding for another bottleneck. For example, shielding link c benefits connecting both (s', d') and (s'', d''). Thus, overlapping bottlenecks should be considered jointly.

Nevertheless, if a set of overlapping bottlenecks do not share common links with another set of overlapping bottlenecks, these two sets can be considered separately, because the shielding decision for one set does not affect the shielding

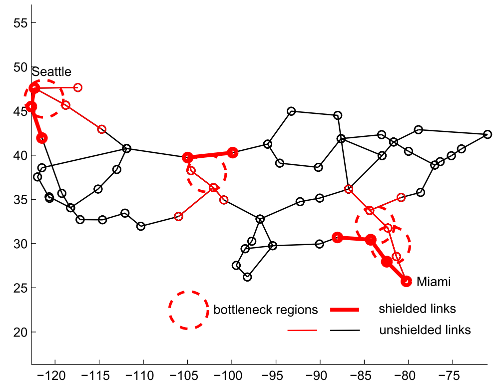


Fig. 12. Bottlenecks and shielded links given disk failure radius 2°.

decision for another in order to shield a path across each bottleneck. The optimal shielding for a set of overlapping bottlenecks is given by MILP (8) which includes constraints only associated with the failures in this set $Z' \subseteq Z$. In short, under the geographical failure model, instead of considering all the failures at once, 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 same network in Fig. 12, where now a failure is any disk with radius 2° longitude/latitude (about 120 miles). Given the SD pair Seattle-Miami, there are four bottleneck regions bounded by dashed circles. The red links inside or intersected by the dashed circles are candidate links to shield. In each of the two disjoint bottlenecks, the shortest path is shielded, illustrated by the thick links with costs 5.9 and 5.1, respectively, while the overlapping bottleneck has shielding cost 10.0, leading to a total shielding cost 21.0. The bottleneck regions and the optimal shielding can both be computed within 1 second.

B. Guaranteeing the Connectivity of a Network

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 removed by a failure event. In order to keep all the nodes connected, one must shield at least a minimum spanning tree.

In the second case, the failure region is tiny and intersects either a single link or all links adjacent to a node. In a graph, a *bridge* is an edge whose removal disconnects the graph. Clearly, every bridge has to be shielded. A *cut node* is a node whose removal disconnects the graph. A graph is *biconnected* if there is no cut node. In a biconnected graph, after removing all the edges adjacent to one node, the other $|V|-1$ nodes are still connected. Therefore, it suffices to guarantee that each node is adjacent to at least one shielded edge, which connects this node with the other nodes after any failure. The optimal set of edges to shield is the *minimum edge cover*, which is a set of edges with the minimum cost such that every node in the graph is adjacent to at least one edge in the set.

A graph may have both bridges and biconnected components (i.e., a subgraph that has no cut node). Since different biconnected components do not share edges, shielding edges in one biconnected component does not benefit another biconnected component. Thus, each biconnected component can be shielded independently by its minimum edge cover. Based on the above development, we have the following algorithm.

Algorithm 3 Optimal Shielding Under the Tiny Disk Failure Model

1) Compute the bridges and biconnected components in a graph.
2) Shield all the bridges. For each biconnected component, shield the edges in its minimum edge cover, where the cost of each edge equals its shielding cost.

Computing bridges and biconnected components takes $O(|E|)$ time [39]. The computation of the minimum edge cover can be reduced to the computation of the maximum matching in a transformed graph in $O(|V|^2)$ time [40], and the computation of the maximum matching takes $O(|V||E|\log|V|)$ time [41]. Thus, the total running time of Algorithm 3 is $O(|V||E|\log|V|)$.

2) Shielding Under the General Failure Model: Now we consider the optimal shielding to guarantee that the network $G(V,E)$ is connected under the general failure model. Each failure $z \in Z$ affects a set of links $E^{(z)}$. Note that it reduces to the geographical failure model if $E^{(z)}$ is a set of links in a geographical region. Our objective is to shield a set of links E^* using the minimum shielding cost, to guarantee that $G(V,E'^{(z)})$ is connected for all $z \in Z$, where $E'^{(z)} = (E \setminus E^{(z)}) \cup E^*$. The variables and parameters have the same representations as those in MILP (8).

$$\min \sum_{(i,j)\in E} c_{ij}h_{ij}/2 \tag{11}$$

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

$$x_{ij}^{(z)sd} - h_{ij} \le 1 - t_{ij}^{(z)} \quad \forall (i,j)\in E, z\in Z, s,d \tag{12}$$

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

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

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

In MILP (11), for each SD pair and failure scenario, there is a flow variable for each link. The number of variables is large because there are many possible failure scenarios. It is difficult to directly solve MILP (11) 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 (12). Given h, it is easy to determine whether there exist feasible flows between all the SD pairs after each failure, by only considering the flow variables and constraints associated with the failure. Benders decomposition can be applied to problems with such separable structure.

Benders decomposition: Benders decomposition accelerates the computation of an optimization problem with separable structure, and has been applied to robust network design [31], [33]. Instead of considering all the constraints at once, it first solves a relaxed problem with a subset of constraints, and then checks whether there are any violated constraints. If there are none, the solution is optimal. Otherwise, a violated constraint is added to the relaxed problem and the problem is solved again. The relaxed problem is called the master problem, and the violated constraints are identified by solving subproblems.

The MILP (11) can be reformulated as follows. It starts with a master problem with constraints only on h.

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

After obtaining h, check whether there are violated constraints by solving subproblems, each corresponding to checking whether the network is connected after a failure. If the linear program (LP) (13) is feasible and has optimal value 0, the network is connected after failure z. If it is infeasible, the associated constraint has been violated.

$$\min 0 \tag{13}$$

$$\text{s.t. } \sum_{\{j|(i,j)\in E\}} x_{ij}^{(z)sd} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(z)sd} = \begin{cases}1, & \text{if } i=s\\ -1, & \text{if } i=d \quad \forall s,d\\ 0, & \text{otherwise}\end{cases} \tag{14}$$

$$x_{ij}^{(z)sd} - h_{ij} \le 1 - t_{ij}^{(z)} \quad \forall (i,j)\in E, s,d \tag{15}$$

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

It is more efficient to add the violated constraint by considering the dual of LP (13). The dual is represented by LP (16), where dual variables p and r correspond to primal constraints (14) and (15), respectively. If LP (16) is unbounded, the constraint $\sum_{sd}[p_d^{*(z)sd} - p_s^{*(z)sd} - \sum_{(i,j)\in E}(1-t_{ij}^{(z)}+h_{ij})r_{ij}^{*(z)sd}] \le 0$ 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 (16) which led to its unboundedness. The added constraints avoid unbounded costs along the extreme rays, to guarantee that LP (16) is bounded and that LP (13) is feasible.

$$\max \sum_{sd}[p_d^{(z)sd} - p_s^{(z)sd} - \sum_{(i,j)\in E}(1-t_{ij}^{(z)}+h_{ij})r_{ij}^{(z)sd}] \tag{16}$$

$$\begin{aligned}\text{s.t. } & p_j^{(z)sd} - p_i^{(z)sd} - r_{ij}^{(z)sd} \le 0 \quad \forall (i,j)\in E, s,d\\ & r_{ij}^{(z)sd} \ge 0 \quad \forall (i,j)\in E, s,d\end{aligned}$$

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. In Benders decomposition, checking whether a subproblem is bounded corresponds to checking whether the network is connected after one failure in the shielding problem. We make the following modification to Benders decomposition. Instead of checking only one failure after obtaining a new shielding decision in each iteration, multiple failures are checked. This is particularly helpful in our problem, because two failures may affect two disjoint sets of links, and the shielding decisions can be made in the same step to reduce the number of iterations. The number of violated constraints added before re-solving the master problem provides a tradeoff between the number of 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% reduction in running time compared with the standard Benders decomposition.

Simulated annealing: Finally, we developed a heuristic based on simulated annealing [42], [43], outlined in Algorithm 4. Simulated annealing is a method to search for globally optimal solutions for nonconvex optimization problems. The search process avoids being stuck in a local minimum without continuing further searches, since a neighbor state with a higher cost can replace the current state with a positive probability.

Algorithm 4 Simulated Annealing for Low-Cost Shielding

Initialization:
Start from an initial state $s = s_0$ where all links are shielded.
Let $c(s)$ denote the shielding cost of s.
Main loop:
for $t = 1$ through $t_{\max}$ **do**
  Compute a feasible neighbor state s' using Algorithm 5.
  Compute the cost difference: $\delta = c(s') - c(s)$.
  if $\delta < 0$ **then**
    $s \leftarrow s'$
  else
    $s \leftarrow s'$ with probability $p = \exp(-\delta/T)$,
    where $T = d/\log(t+1)$ and $d > 0$.
  end if
end for

During the first fewer iterations, the *temperature* T is large so that it is easy to enter a state that has a larger shielding cost to explore more possible states. As the number of iterations increases, T decreases and makes it less likely to enter a state with a larger shielding cost. If the number of iterations is large, T approaches 0 and the algorithm terminates at a state which has the smallest shielding cost compared with all its neighbors. As suggested by [43], T is set according to $T(t) = d/\log\ (t+1)$, where t is the number of the current iteration and d is a positive constant. After tuning the parameters, we set $d = 10$ and $t_{\max} = 200$ in our algorithm to compute the shielding in the XO network.

The neighbor states in the iterations of simulated annealing are computed by Algorithm 5. The probability of removing a link from the shielded set is proportional to the shielding cost of the link. Thus, links with 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

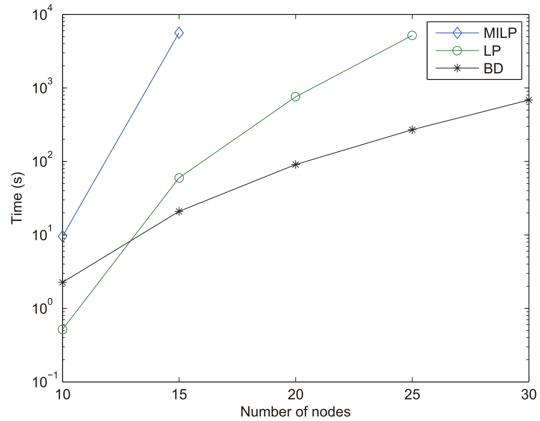


Fig. 13. Running time comparisons of solving Benders decomposition, directly solving MILP and its LP relaxation.

shielding cost, so that links with smaller shielding costs are more likely to be added.

Algorithm 5 Computing a Feasible Neighbor State

Let S be the set of shielded links and S^c be the set of unshielded links in the current state. Perform one of the following operations.

1) Randomly move a link from S to S^c.
2) Randomly move a link l from S to S^c, and randomly move a link (except l) from S^c to S.
3) Randomly move a link from S^c to S.

Since the objective is to find a neighbor state with a smaller shielding cost, the operations are done sequentially during the first few iterations. For example, after one shielded link becomes unshielded (operation 1), if the current shielding is feasible, a neighbor state with a smaller shielding cost is identified. If any removal of shielded link from the shielded set 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 obtain a new feasible shielding state.

If the three operations are always performed sequentially, the algorithm may end up in cycles. For example, it is possible that both operations 1 and 2 cannot find a feasible shielding state, and a possible neighbor state is to shield an extra link. Starting from the neighbor state, the only link that can be removed without causing infeasible shielding is the link that was just added. Thus, after some iterations when many redundant links are removed from the shielded set, the operations are done randomly to avoid such cycles.

3) Numerical Results: We first compare the running time of solving the MILP that computes the optimal shielding to guarantee network connectivity using Benders decomposition, solving the MILP directly, and solving its LP relaxation. Fig. 13 illustrates the growth of the running time of these solutions as network size increases. The results are averaged over 10 instances of connected random graphs, where an edge exists with a given probability between all pairs of nodes. The number of nodes of the graph is varied from 10 to 30, with average degree 5. We consider failures of unshielded links adjacent to any pair of nodes. 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 on the shielding costs. Solving the MILP using Benders decomposition is faster than directly solving the LP relaxation for larger networks.

TABLE II

RUNNING TIME COMPARISONS OF SA, BD AND THE MODIFIED BD FOR RANDOM GRAPHS

Number of Nodes	Average 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

TABLE III

SHIELDING COST COMPARISONS OF SA RESULTS AND EXACT SOLUTIONS FOR RANDOM GRAPHS

Number of Nodes	Average Degree	SA Cost	Optimal 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 IV

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°	61.75	109.19	106.1	99.5	0.066
2°	70.75	875.12	124.6	121.3	0.027

Moreover, the modified Benders decomposition (adding multiple violated constraints in each iteration) reduces the running time further by more than 50% in most cases as shown in the last column of Table II.

Next we compare the performance of simulated annealing with the modified Benders decomposition. The cost of shielding an edge equals the sum of the indices (from 1 to $|V|$) of its end nodes in the randomly generated graphs. We observe in Tables II and III 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 $3\% \sim 6\%$.

Finally we apply our algorithms to obtain the optimal shielding for the XO network. Fig. 14 illustrates the optimal shielding to guarantee the connectivity of the entire XO backbone network after any disk failure with radius 1°.

Simulated annealing also has good performance in solving the shielding problem for the XO network. The results are shown in Table IV. Thus, simulated annealing can be used to solve larger size problems when the computation time to obtain the exact solutions is too long.

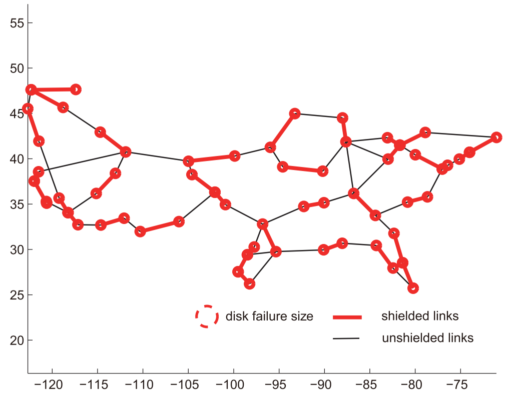


Fig. 14. Optimal shielding in the XO network to guarantee network connectivity after any disk failure with radius 1°.

IV. Guaranteeing Partial Connectivity of a Network

In most networks, a large number of links need to be shielded to guarantee the full connectivity of the network. In fact, even in the tiny disk failure case where the network is biconnected, all the links in the minimum edge cover need to be shielded. The number of shielded links is at least half the number of nodes. For larger failures, or if the network has a tree structure, even more links need to be shielded.

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

We compute the minimum cost 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 connected SD pairs after a failure. Compared with MILP (11), the flow constraints are not imposed to every SD pair, but a fraction of SD pairs. In constraints (18), $I^{(z)sd}$ can be either 0 or 1, where $I^{(z)sd} = 1$ if (s, d) are connected under failure z. The total number of connected SD pairs should be at least a fraction α of all the $N(N-1)/2$ SD pairs, guaranteed by constraints (19), where α is the ATTR requirement and $N = |V|$ is the total number of nodes.

$$\min \sum_{(i,j)\in E} c_{ij} h_{ij}/2 \tag{17}$$

$$\text{s.t.} \sum_{\{j|(i,j)\in E\}} x_{ij}^{(z)sd} - \sum_{\{j|(j,i)\in E\}} x_{ji}^{(z)sd} = \begin{cases} I^{(z)sd}, & \text{if } i = s \\ -I^{(z)sd}, & \text{if } i = d \\ 0, & \text{otherwise} \end{cases} \quad \forall z \in Z, s, d \tag{18}$$

$$\begin{aligned} & x_{ij}^{(z)sd} - h_{ij} \le 1 - t_{ij}^{(z)} \quad \forall (i,j) \in E, z \in Z, s, d \\ & \sum_{sd} I^{(z)sd} \ge \alpha N(N-1)/2 \quad \forall z \in Z \end{aligned} \tag{19}$$

$$\begin{aligned} & h_{ij} - h_{ji} = 0 \quad \forall (i,j) \in E \\ & x_{ij}^{(z)sd} \ge 0 \quad \forall (i,j) \in E, z \in Z, s, d \\ & h_{ij} = \{0,1\} \quad \forall (i,j) \in E \\ & I^{(z)sd} = \{0,1\} \quad \forall z \in Z, s, d \end{aligned}$$

The above MILP (17) for partial connectivity has more variables and constraints than MILP (11) for full connectivity. First, there are the additional variables $I^{(z)sd}$. Moreover, in MILP (11), checking the connectivity between node 1 and nodes $2, 3, \ldots, N$ is enough to check the connectivity of the entire network. In contrast, in MILP (17), $N(N-1)/2$ SD pairs need to be checked. Thus, the running time of MILP (17) is larger than the running time of MILP (11). Even when $\alpha = 1$ and the two MILPs have the same optimal solution, the running time of MILP (17) can be 20 times longer than the running time of MILP (11) for a network that has 10 nodes.

Moreover, the running time of MILP (17) depends on α. Smaller α leads to a larger feasibility region, which makes it harder to check the optimality of the solution and requires a longer running time (unless α is very small so that no or few links need to be shielded). For example, for a network that has 10 nodes, if one node is disconnected from the others, the ATTR is $(9 \times 8)/(10 \times 9) = 0.8$. Thus, the MILPs in which $\alpha = 0.9$ and $\alpha = 1$ have the same optimal solution (i.e., the optimal shielding that guarantees network full connectivity). The running time of the MILP in which $\alpha = 1$ is 4 minutes while the running time of the MILP in which $\alpha = 0.9$ is more than one hour.

In contrast with the full connectivity case, Benders decomposition for MILP (17) does not significantly reduce its running time. In Benders decomposition, the constraints added to the master problem progressively cut the feasibility region of the relaxed master problem to approach the feasibility region of the original problem. In the partial connectivity case, a fraction of SD pairs are allowed to be disconnected. The added constraints to the master problem do not reduce the feasibility region of the relaxed problem as effective as the added constraints in the full connectivity case. Thus, the Benders decomposition for partial connectivity is not as effective in reducing the running time, and is omitted.

On the other hand, the simulated annealing in the previous section can be used for partial network connectivity with a small modification. The only difference is in determining whether the shielding is feasible when computing a neighbor state. In the simulated annealing for partial connectivity, as long as the ATTR is above α, the shielding is feasible and is a candidate for the next state. Checking whether a graph is connected and checking whether the ATTR is above a given value can both be done in $O(|E|)$ time using a depth-first-search and computing the sizes of the connected components. Since a feasible shielding for partial connectivity may not be feasible for full connectivity, fewer trials and less computation are needed to find a feasible neighbor state for partial connectivity. To conclude, the running time per iteration in the

TABLE V

COST REDUCTION FOR PARTIAL CONNECTIVITY OF THE XO NETWORK

Attack radius	Optimal Cost (full connectivity)	SA Cost ($\alpha = 96\%$)	SA Cost ($\alpha = 93\%$)
1°	99.5	26.5	5.0
2°	121.3	81.2	49.7

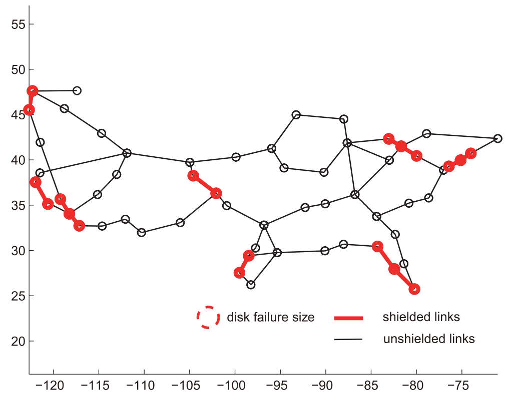


Fig. 15. A feasible shielding for the XO network to guarantee that $\alpha = 96\%$ after any disk failure with radius 1°.

simulated annealing for partial network connectivity is slightly smaller compared with that for full network connectivity.

In our numerical tests, we were able to solve the partial connectivity problem using MILP on small networks which have no more than 15 nodes. In contrast, simulated annealing that solves the partial connectivity problem has shorter running time compared with the full connectivity case. (The reduction in running time is larger for smaller α, and varies from 10% to 50% in our tests.) Moreover, the shielding costs obtained by the simulated annealing are nearly identical with the exact solutions for different levels of ATTR requirement in our tests.

The numerical results suggest that the shielding cost is significantly reduced by relaxing the connectivity constraint to allow for values of α less than one. If one node is allowed to be disconnected in the XO network, $\alpha = (59 \times 58)/(60 \times 59) > 0.96$, and a feasible shielding is depicted in Fig. 15 if the disk failure has radius 1°. Much fewer links are shielded compared with Fig. 14 that guarantees the full connectivity. Table V suggests that the cost reduction is larger for smaller failure. Under disk failure of radius 1°, the shielding cost is reduced by at least 73% if one node is allowed to be disconnected, and is reduced by 95% if two nodes are allowed to be disconnected, in which case $\alpha = (58 \times 57)/(60 \times 59) > 93\%$. Under disk failure of radius 2°, the shielding costs are reduced by at least 33% and 59%, if one or two nodes are allowed to be disconnected, respectively.

V. CONCLUSION

In this paper, we developed theory and algorithms for network shielding to improve network robustness. We exploit the connections between the shielding problem and graph augmentation problems as well as the fixed-charge problems, and developed algorithms and MILPs to obtain the minimum cost shielding to increase the edge-connectivity. The solutions can be used to improve the network reliability under independent random link failures.

We developed MILP formulations for the optimal shielding to guarantee the connectivity of a single SD pair and the network under a general failure model. To guarantee the connectivity of a single SD pair under a geographical failure model, we decompose the problem to multiple subproblems, each of which computes the optimal shielding for links in a geographical region. The MILP that guarantees network connectivity has separable structure, for which Benders decomposition can be applied to reduce the running time by over two orders of magnitude. A slightly modified Benders decomposition further reduces the running time by more than 50%. In addition, we used simulated annealing to obtain near-optimal solutions in much shorter running time.

Much less shielding cost is required to guarantee partial connectivity of a network, even in the case where at most one node is allowed to be disconnected. We observe larger reduction in the shielding cost if the size of a geographical failure region is small. The algorithms can be easily modified to solve the problem that guarantees the connectivity of a selected set of SD pairs. For example, in the MILP, the flow constraints can be imposed only for the selected SD pairs. In the simulated annealing, the connectivity of the selected SD pairs can be checked to determine the feasibility of shielding, while the remaining algorithm is unchanged.

In summary, the methods in this paper can be used to construct new networks and upgrade existing networks to increase network reliability under random failures, and to improve network robustness under geographical and general failures.

REFERENCES

[1] J. S. Foster, Jr., *et al.*, "Report of the commission to assess the threat to the United States from electromagnetic pulse (EMP) attack: Critical national infrastructures," EMP Commission, McLean, VA, USA, Tech. Rep., Apr. 2008. [Online]. Available: http://www.empcommission.org/docs/A2473-EMP_Commission-7MB.pdf

[2] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of the fiber infrastructure to disasters," *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1610–1623, Dec. 2011.

[3] 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.*, vol. 21, no. 5, pp. 1525–1538, Oct. 2013.

[4] S. Neumayer, A. Efrat, and E. Modiano, "Geographic max-flow and min-cut under a circular disk failure model," *Comput. Netw.*, vol. 77, pp. 117–127, Feb. 2015.

[5] C. Cao, M. Zukerman, W. Wu, J. H. Manton, and B. Moran, "Survivable topology design of submarine networks," *J. Lightw. Technol.*, vol. 31, no. 5, pp. 715–730, Mar. 1, 2013.

[6] S. Borne, E. Gourdin, B. Liau, and A. Mahjoub, "Design of survivable IP-over-optical networks," *Ann. Oper. Res.*, vol. 146, no. 1, pp. 41–73, 2006.

[7] D. Papadimitriou *et al.*, "Inference of shared risk link groups," Working Draft, IETF Secretariat, Internet-Draft, Tech. Rep. IETF, Nov. 2001. [Online]. Available: http://draft-many-inference-srlg-02.txt

[8] H. Kerivin and A. R. Mahjoub, "Design of survivable networks: A survey," *Networks*, vol. 46, no. 1, pp. 1–21, Aug. 2005.

[9] D. R. Karger, "A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem," *SIAM Rev.*, vol. 43, no. 3, pp. 499–522, 2001.

[10] H.-W. Lee, K. Lee, and E. Modiano, "Maximizing reliability in WDM networks through lightpath routing," *IEEE/ACM Trans. Netw.*, vol. 22, no. 4, pp. 1052–1066, Aug. 2014.

[11] G. Maier, A. Pattavina, S. De Patre, and M. Martinelli, "Optical network survivability: Protection techniques in the WDM layer," *Photon. Netw. Commun.*, vol. 4, no. 3, pp. 251–269, 2002.

[12] M. Medard, D. Marquis, R. A. Barry, and S. G. Finn, "Security issues in all-optical networks," *IEEE Netw.*, vol. 11, no. 3, pp. 42–48, May 1997.

[13] T. Watanabe and A. Nakamura, "Edge-connectivity augmentation problems," *J. Comput. Syst. Sci.*, vol. 35, no. 1, pp. 96–144, Aug. 1987.

[14] D. Naor, D. Gusfield, and C. Martel, "A fast algorithm for optimally increasing the edge connectivity," *SIAM J. Comput.*, vol. 26, no. 4, pp. 1139–1165, 1997.

[15] S. Khuller and U. Vishkin, "Biconnectivity approximations and graph carvings," *J. ACM*, vol. 41, no. 2, pp. 214–235, 1994.

[16] T. Mashima and T. Watanabe, "Approximation algorithms for the k-edge-connectivity augmentation problem," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 1. Seattle, WA, USA, Apr. 1995, pp. 155–158.

[17] *Electromagnetic Pulse Survivability of Telecommunication Assets*, Booz Allen Hamilton, McLean, VA, USA, Feb. 1987.

[18] B. Overton *et al.*, "Microbend-resistant optical fiber," U.S. Patent 8 145 027 B2, Mar. 27, 2012.

[19] O. Tatat, "Optical fiber telecommunications cable," U.S. Patent 7 646 954 B2, Jan. 12, 2010.

[20] S. Gibbs. (Aug. 14, 2014). *Google Reinforces Undersea Cables After Shark Bites*. [Online]. Available: http://gu.com/p/4vm78/stw

[21] C. Liu, Y. Fan, and F. Ordóñez, "A two-stage stochastic programming model for transportation network protection," *Comput. Oper. Res.*, vol. 36, no. 5, pp. 1582–1590, 2009.

[22] L. V. Snyder, M. P. Scaparra, M. S. Daskin, and R. L. Church, "Planning for disruptions in supply chain networks," in *Proc. Tuts. Oper. Res. (INFORMS)*, 2006, pp. 234–257.

[23] R. L. Church, M. P. Scaparra, and R. S. Middleton, "Identifying critical infrastructure: The median and covering facility interdiction problems," *Ann. Assoc. Amer. Geograph.*, vol. 94, no. 3, pp. 491–502, 2014.

[24] G. Brown, M. Carlyle, J. Salmerón, and K. Wood, "Defending critical infrastructure," *Interfaces*, vol. 36, no. 6, pp. 530–544, Nov. 2006.

[25] M. Dziubiński and S. Goyal, "Network design and defence," *Games Econ. Behavior*, vol. 79, pp. 30–43, May 2013.

[26] W. H. Cunningham, "Optimal attack and reinforcement of a network," *J. ACM*, vol. 32, no. 3, pp. 549–561, Jul. 1985.

[27] A. Alashaikh, T. Gomes, and D. Tipper, "The spine concept for improving network availability," *Comput. Netw.*, vol. 82, pp. 4–19, May 2015.

[28] J. Zhang, E. Modiano, and D. Hay, "Enhancing network robustness via shielding," in *Proc. 11th Int. Conf. Design Rel. Commun. Netw. (DRCN)*, Kansas City, MO, USA, Mar. 2015, pp. 17–24.

[29] T. Jordán, "Two NP-complete augmentation problems," Odense University, Odense, Denmark, Tech. Rep., 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.9908&rep=rep1&type=pdf

[30] P. Gray, "Technical note—Exact solution of the fixed-charge transportation problem," *Oper. Res.*, vol. 19, no. 6, pp, pp. 1529–1538, 1971.

[31] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. San Mateo, CA, USA: Morgan Kaufmann, 2004.

[32] D. Bienstock and N. Diaz, "Blocking small cuts in a network, and related problems," *SIAM J. Comput.*, vol. 22, no. 3, pp. 482–499, 1993.

[33] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA, USA: Athena Scientific, 1997.

[34] C.-L. Li, "The point-to-point delivery and connection problems: Complexity and algorithms," *Discrete Appl. Math.*, vol. 36, no. 3, pp. 267–292, 1992.

[35] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, no. 2, pp. 248–264, 1972.

[36] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987.

[37] XO Communications. (Apr. 2015). *Network Map*. http://www.xo.com/about/network/Pages/maps.aspx

[38] D. R. Karger and C. Stein, "A new approach to the minimum cut problem," *J. ACM*, vol. 43, no. 4, pp. 601–640, 1996.

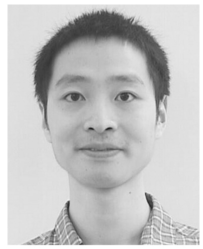
[39] J. M. Schmidt, "A simple test on 2-vertex-and 2-edge-connectivity," *Inf. Process. Lett.*, vol. 113, no. 7, pp. 241–244, 2013.

[40] J. Plesnik, "Equivalence between the minimum covering problem and the maximum matching problem," *Discrete Math.*, vol. 49, no. 3, pp. 315–317, 1984.

[41] Z. Galil, S. Micali, and H. Gabow, "An $O(EV \log V)$ algorithm for finding a maximal weighted matching in general graphs," *SIAM J. Comput.*, vol. 15, no. 1, pp. 120–130, 1986.

[42] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statist. Sci.*, vol. 8, no. 1, pp. 10–15, 1993.

[43] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, vol. 13, no. 2, pp. 311–329, 1988.



Jianan Zhang received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012, and the M.S. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2014, where he is currently pursuing the Ph.D. degree with the Laboratory for Information and Decision Systems. His current research interests include network robustness, optimization, and interdependent networks.

Eytan Modiano received the B.S. degree in electrical engineering and computer science from the University of Connecticut, Storrs, CT, USA, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, USA, in 1989 and 1992, respectively. He was a Naval Research Laboratory Fellow from 1987 to 1992 and a National Research Council Post-Doctoral Fellow from 1992 to 1993. From 1993 to 1999, he was with the Lincoln Laboratory, Massachusetts Institute of Technology (MIT). Since 1999, he has been on the faculty with MIT, where he is currently a Professor with the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems. His research interests include communication networks and protocols with emphasis on satellite, wireless, and optical networks. He was a co-recipient of the MobHoc 2016 Best Paper Award, the Wiopt 2013 Best Paper Award, and the Sigmetrics 2006 Best Paper Award. He is an Editor-at-Large of the IEEE/ACM TRANSACTIONS ON NETWORKING, and served as an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and the IEEE/ACM TRANSACTIONS ON NETWORKING. He was the Technical Program Co-Chair of the IEEE Wiopt 2006, the IEEE Infocom 2007, the ACM MobiHoc 2007, and the DRCN 2015. He is a Fellow of the IEEE and an Associate Fellow of the AIAA, and served on the IEEE Fellows Committee.

David Hay received the B.A. (*summa cum laude*) and Ph.D. degrees in computer science from the Technion–Israel Institute of Technology, Haifa, Israel, in 2001 and 2007, respectively. He was with the IBM Haifa Research Labs, Haifa; Cisco Systems, San Jose, CA, USA; the Electronic Department, Politecnico di Torino, Turin, Italy; and the Electrical Engineering Department, Columbia University, New York, NY, USA. In 2010, he co-founded (with Prof. Brembler-Barr) the DEEPNESS Laboratory, where he was involved in deep packet inspection in next-generation network devices. He is currently an Associate Professor with the Rachel and Selim Benin School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel. His research interests include computer networks in particular, network algorithmics, packet classification, deep packet inspection, network survivability, and software-defined networking.