

Providing Protection in Multi-Hop Wireless Networks

Greg Kuperman
MIT LIDS
Cambridge, MA 02139
gregk@mit.edu

Eytan Modiano
MIT LIDS
Cambridge, MA 02139
modiano@mit.edu

Abstract—We consider the problem of providing protection against failures in wireless networks subject to interference constraints. Typically, protection in wired networks is provided through the provisioning of backup paths. This approach has not been previously considered in the wireless setting due to the prohibitive cost of backup capacity. However, we show that in the presence of interference, protection can often be provided with no loss in throughput. This is due to the fact that after a failure, links that previously interfered with the failed link can be activated, thus leading to a “recapturing” of some of the lost capacity.

We provide both an ILP formulation for the optimal solution, as well as algorithms that perform close to optimal. More importantly, we show that providing protection in a wireless network uses as much as 72% less protection resources as compared to similar protection schemes designed for wired networks, and that in many cases, no additional resources for protection are needed.

I. INTRODUCTION

Multi-hop wireless mesh networks have become increasingly ubiquitous, with wide-ranging applications from military to sensor networks. As these networks continue gaining in prominence, there is an increasing need to provide protection against node and link failures. In particular, wireless mesh networks have recently emerged as a promising solution for providing Internet access. Since these networks will be tightly coupled with the wired Internet to provide Internet services to end-users, they must be equally reliable. Wired networks have long provided pre-planned backup paths, which offer rapid and guaranteed recovery from failures. These protection techniques cannot be directly applied to wireless networks due to interference constraints. As opposed to wired networks, two wireless nodes in close proximity will interfere with one another if they transmit simultaneously in the same frequency channel. So, in addition to finding a backup route, a schedule of link transmissions needs to be specified. In this work, we consider the problem of providing guaranteed protection in wireless networks with interference constraints via pre-planned backup routes, as well as their corresponding link transmission schedules.

Guaranteed protection schemes for wired networks have been studied extensively [1–5], with the most common scheme being

This work was supported by NSF grants CNS-1116209 and CNS-0830961, by DTRA grant HDTRA-09-1-005, and by the Department of the Air Force under Air Force contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

1 + 1 guaranteed path protection [5]. The 1 + 1 protection scheme provides an edge-disjoint backup path for each working path, and guarantees the full demand to be available at all times after any single link failure. Protection schemes optimized for wireless networks with interference constraints have not yet been considered. Typically, an approach for resiliency in wireless networks (in particular sensor networks) is to ensure that there exists “coverage” for all nodes given some set of link failures [6, 7]. This approach to resiliency does not consider routing and scheduling with respect to interference constraints, and assumes that there exists some mechanism to find a route and schedule at any given point in time. Furthermore, there is no guarantee that sufficient capacity will be available to protect against a failure. The idea of applying 1 + 1 protection in wireless networks is briefly mentioned in [8]. However, [8] does not study the specific technical details of such an approach to wireless protection. The goal of this paper is to study protection mechanisms for wireless networks with a particular focus on the impact of wireless interference and the need for scheduling.

The addition of interference constraints makes the protection problem in a wireless setting fundamentally different from the ones found in a wired context. After a failure in a wireless network, links that could not have been used due to interference with the failed link become available, and can be used to recover from the failure. In fact, it is often possible to add protection in a wireless setting without using any additional resources.

Consider allocating a protection route for the following example, shown in Fig. 1. The wireless network operates in a time-slotted fashion, with equal length time slots available for transmission. Any two nodes within transmission range have a link between them, and each link’s time slot assignment is shown in the figures. We assume a 1-hop interference model where any two links that have a node in common cannot be active at the same time. Additionally, we assume unit capacity links. Before any failure, the maximum flow from s to d is 1, and can be achieved using a two time slot schedule, as shown in Fig. 1a. At any given point in time, only one outgoing link from s can be active, and similarly, only one incoming link to d can be active. Wireless links $\{s, c\}$, and $\{c, d\}$ cannot be used prior to the failure of $\{s, b\}$, but become available after $\{s, b\}$ fails. After the failure of $\{s, b\}$, flow can be routed from s to c during time slot 2, and from c to d during slot 1, as shown in Fig. 1b.

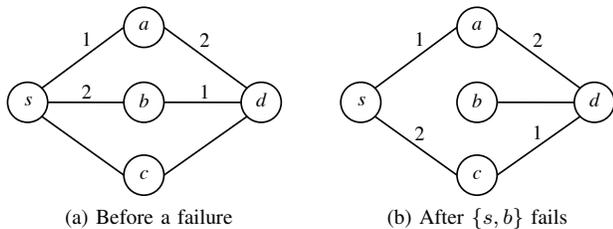


Fig. 1: Time slot assignment for protection in a wireless network

Similar schedules can be found for failures of the other links. The maximum flow from s to d is 1 for both before and after a failure; i.e., there is no reduction in maximum throughput when allocating resources for a protection route on $\{s, c\}$ and $\{c, d\}$: protection can be assigned for “free”. This is in contrast to a wired network where the maximum throughput without protection from s to d is 3, and the maximum throughput when assigning a protection route on $\{s, c\}$ and $\{c, d\}$ is 2, which amounts to a $\frac{1}{3}$ loss in throughput due to protection.

The novel contributions of this paper is introducing the Wireless Guaranteed Protection (WGP) problem in multi-hop networks with interference constraints. In Section II, the model for WGP is presented. In Section III, properties of an optimal solution are presented for a single demand with 1-hop interference constraints, which are then used to motivate the development of a polynomial time optimal algorithm. In Section IV, an optimal solution is developed via a mixed integer linear program for general interference constraints. In Section V, time-efficient algorithms are developed that perform within 4.5% of the optimal solution.

II. MODEL AND PROBLEM DESCRIPTION

In this paper, solutions to the guaranteed protection problem for multi-hop wireless networks subject to interference constraints are developed and analyzed. Our goal is to provide protection in a manner similar to what has been done in the wired setting. Namely, after the failure of some network element, all connections must maintain the same level of flow that they had before the failure. In order to do so, resources are allocated and scheduled in advance on alternate (backup) routes to protect against failures.

In wired networks, two adjacent nodes can transmit simultaneously because they do not interfere with one another; if capacity exists on a set of links, a path can be routed using that capacity. Wireless networks are different; interference constraints must be considered. A set of links in close proximity cannot transmit simultaneously on the same frequency channel; only one link from that set can be active at a time, or else they will interfere with one another. Not only must a path between the source and destination be found with available capacity, but also a schedule of link transmissions needs to be determined. This is known as the routing and scheduling problem [8–16], which is known to be NP-Hard [9].

The addition of interference constraints adds complexity to the traditional wired protection problem, but also presents an opportunity to gain protection from failures with minimal loss

of throughput. After a failure in a wireless network, links that could not have been used due to interference with the failed link become available, and can be used to recover from the failure. In fact, it is often possible to add protection in a wireless setting without any loss in throughput.

The following network model is used for the remainder of the paper. A graph G has a set of vertices V and edges E . An interference matrix \mathcal{I} is given, where $I_{ij}^{kl} \in \mathcal{I}$ is 1 if links $\{i, j\}$ and $\{k, l\}$ can be activated simultaneously (do not interfere with each other), and 0 otherwise. The interference matrix is agnostic to the interference model used (i.e., it can be used to represent nearly any type of link interferences). For the remainder of this work, we focus on the 1-hop interference model (any two links that share a node cannot be activated simultaneously), but our schemes can be adapted to the K -hop [17] interference model as well. Our goal in this paper is to develop a framework for routing and scheduling with protection under interference constraints. We assume nodes are fixed, links are bidirectional, and that the network uses a synchronous time slotted system, with equal length time slots; the set of time slots used is \mathcal{T} . Only link failures are considered, and a single-link failure model is assumed; it is straightforward to apply the solutions developed in this paper to node failures as well. For now, we assume centralized control; the algorithms presented can be modified to work in a distributed fashion, as done in [18]. Additionally, we only consider a single frequency channel.

III. OPTIMAL SOLUTION FOR A SINGLE DEMAND

In this section, we aim to achieve insight into providing protection for wireless networks with interference constraints by examining the optimal solution for a single demand under a basic set of network parameters: 1-hop interference constraints and unit capacity links. In Section III-A, properties of an optimal solution are shown for routing and scheduling with and without protection. In Section III-B, a polynomial time optimal algorithm is developed using the properties found. All proofs in this section are omitted for brevity, and can be found in [19].

A. Properties of an Optimal Solution

In this section, properties of an optimal solution for WGP for a single demand are found. First, routing and scheduling without protection is examined, and then the results are extended to the protection setting.

Observation 1. *The maximum flow that can be routed and scheduled between the source s and destination d under 1-hop interference constraints without protection is 1.*

The intuition behind Observation 1 is that since at most one unit-capacity edge leaving the source can be active during any given time slot, the maximum flow that can leave the source is 1. While this indicates that a flow of 1 is possible, it does not necessarily mean that a flow of 1 can be achieved. We now give the properties of maximum flows in a wireless network under 1-hop interference constraints.

Lemma 1. *To achieve the maximum flow of 1, there must exist at least two node-disjoint paths from s to d .*

Corollary 1. *If two node-disjoint paths from s to d do not exist, then the maximum flow is $\frac{1}{2}$.*

Any path from s to d can have an interference-free schedule by alternating between time slots 1 and 2 for each edge of the path; hence, any edge of the path will only be active for half of the time, and the path will support a flow of $\frac{1}{2}$. If two or more node-disjoint paths do exist, then a maximum flow of 1 or $\frac{2}{3}$ is obtainable depending on the total number of edges in the disjoint paths.

Lemma 2. *If there exists two node-disjoint paths between s and d with an even total number of edges over both paths, then the maximum flow of 1 is achievable. If there exists only two node-disjoint paths with an odd total number of edges over both paths, then the maximum flow is $\frac{2}{3}$.*

To help see Lemma 2, three simple examples are shown in Fig. 2, with the time slot assignments for the links shown in the figures. In Fig. 2a and 2b, there are two node-disjoint paths from the source s to destination d that have an even total number of edges. In Fig. 2a, each path has an even number of edges, and in Fig. 2b, each path has an odd number of edges. An interference-free schedule for the two paths can be found using two time slots. Each link is active for $\frac{1}{2}$ of the time; hence, each path can support a flow of $\frac{1}{2}$, giving a total flow of 1. In Fig. 2c, the two node-disjoint paths have an odd total number of edges; three time slots are needed to schedule the two paths. Each link is active for $\frac{1}{3}$ of the time, and each path can support a flow of $\frac{1}{3}$, which gives a total flow of $\frac{2}{3}$.

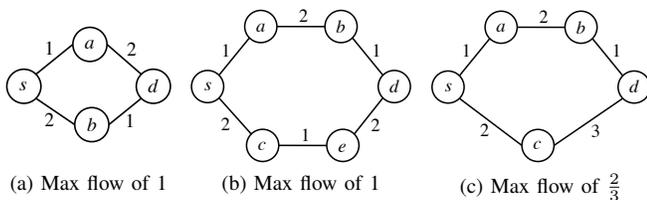


Fig. 2: Node-disjoint paths with an even or odd total number of edges

Corollary 2. *If there exists more than two node-disjoint paths between s and d , a maximum flow of 1 is always achievable.*

If there exists more than two node-disjoint paths, then there always exists a pair of node-disjoint paths with an even number of edges.

These results can be extended to the case where protection is required. For protection against any single link failure in a graph $G = (V, E)$, consider each subgraph after a link failure: $G^e = (V, E \setminus e)$, $e \in E$; all the previous results still apply to each of these new subgraphs. To find the maximum possible protected flow, the maximum flow is found after each edge is individually removed (each possible edge failure). The *minimum* of these flows is the maximum protected flow.

B. Time-Efficient Optimal Algorithm

With properties of an optimal solution to offer guaranteed protection for a single demand under 1-hop interference constraints established, it is now possible to construct an optimal polynomial timed algorithm. There are three possible maximum

flows between nodes s and d , which follow from the results in Section III-A:

- 1 if there exists two node-disjoint paths with an even total number of edges.
- $\frac{2}{3}$ if there exists two node-disjoint paths with an odd total number of edges.
- $\frac{1}{2}$ if there are no node-disjoint paths.

To find the maximum protected flow between nodes s and d in a graph $G = (V, E)$, the maximum flow is found for each link failure by using a subgraph with each link e removed: $G^e = (V, E \setminus e)$, $e \in E$. The minimum of these maximum flows is the maximum protected flow possible for the demand.

The key to finding the maximum protected flow is to be able to identify node-disjoint paths between s and d with either an even or odd total number of edges. If there are at most two node-disjoint paths, then the maximum flow can only be found if it is possible to find a pair of paths with an even total number of edges. Hence, we focus on trying to find a pair of node-disjoint paths that have an even total number of edges over both of the paths. There has been limited work on trying to identify shortest paths with an even number of edges [20], but no work looking at such an algorithm for disjoint paths.

Development of the optimal algorithm is as follows: we first find the shortest pair of *edge*-disjoint paths with an even number of total edges, and then we extend this algorithm to find the shortest pair of *node*-disjoint paths with an even number of total edges.

1) *Shortest pair of edge-disjoint paths with an even number of total edges:* To find the shortest pair of edge-disjoint paths with an even number of edges, we begin by considering the more general case without the even-edge restriction (the paths can have any number of edges), which was previously considered in [21]. We use a different formulation for the problem by using *minimum-cost flows*, which are defined as finding a flow of minimum cost between a source and destination in a network that has both edge costs and edge capacities [22]. Minimum-cost flows have the property that when given all integer inputs (for edge costs and capacities), they will have all integer solutions (integer flows). We solve the shortest disjoint pair of paths problem by solving the following optimization problem: find a flow of minimum cost to route two units from s to d in a graph with unit capacity and unit cost edges. This will find the shortest pair of disjoint paths since two units of flow need to be routed, no edge can have more than a single unit of flow, and with integer inputs, the solution will be integer, which will be two edge-disjoint paths of unit flow and minimum cost.

One algorithm to solve the minimum-cost flow problem is the successive shortest paths (SSP) algorithm [22]. SSP finds the shortest path, and routes the maximum flow possible onto that path. This repeats until the desired flow between the source and destination is routed. SSP runs in polynomial time to solve the minimum-cost flow formulation for the shortest pair of disjoint paths; further details of SSP can be found in [22].

Using SSP to solve for a minimum-cost flow requires the use of some shortest path algorithm. Assume there exists a shortest

path algorithm that is capable of finding a path with an even or odd number of edges; label these algorithms Even and Odd Shortest Path (ESP and OSP, respectively). Using SSP to solve for the shortest pair of disjoint paths with either ESP or OSP as the shortest path function will always yield a pair of disjoint paths with an even total number of edges (if they exist). We call this the Even Shortest Pair of Edge-Disjoint Paths algorithm.

Lemma 3. *The Even Shortest Pair of Edge-Disjoint Paths algorithm will find, if it exists, the shortest pair of disjoint paths with an even total number of edges.*

In order to use SSP to find the shortest pair of disjoint paths with an even number of edges, a shortest path algorithm is needed that can find a path with an even or odd number of edges. The algorithm in [20] that finds the shortest path with an even number of edges cannot be easily extended to find the shortest pair of disjoint paths with an even number of edges. Hence, we first focus on developing an algorithm to find the Even Shortest Path (ESP), and then extend ESP to find the Odd Shortest Path (OSP).

We modify the standard Bellman-Ford recursion [22] to search for only paths with an even number of edges, which is shown in Equation 1. We label $S_z(s, k)$ to be the minimum-cost path from node s to k using at most $2z$ edges. The cost of edge $\{i, j\}$ is c_{ij} ; in our case $c_{ij} = 1, \forall \{i, j\} \in E$. Instead of checking if a path from s to j plus edge $\{j, k\}$ is of lower cost than the existing path from s to k , we check to see if the path from s to i plus two edges $\{i, j\}$ and $\{j, k\}$ are of lower cost than the existing path from s to k .

$$S_z(s, k) = \min \left[\min_{\substack{\{i, j\} \in E \\ \{j, k\} \in E \\ i \neq k}} (S_{z-1}(s, j) + c_{ij} + c_{jk}), S_{z-1}(s, k) \right], \\ \forall z = 1..|V|, \forall k \in V \quad (1)$$

To find the shortest path from the source s with an odd number of edges, we run ESP from all neighboring nodes of s (nodes that are one hop from s). The lowest cost path leading back to the source is the solution to OSP.

2) *Shortest pair of node-disjoint paths with an even number of total edges:* In order to optimally solve for routing and scheduling under 1-hop interference constraints, a pair of node-disjoint paths with an even number of edges must be found. The Even Shortest Pair of Edge-Disjoint Paths algorithm finds the shortest pair of edge-disjoint paths with an even number of edges. To use the edge-disjoint algorithm to solve the node-disjoint case, each node is transformed into two separate nodes with an edge of zero cost between them: one node has all incoming edges, and the other all outgoing (as shown in Fig. 3). If there existed multiple edge-disjoint paths that intersected at node v , they would no longer be able to be edge-disjoint in the transformed network, because then they would all have to share the edge $\{v_{in}, v_{out}\}$.

Running the Even Shortest Pair of Edge-Disjoint Paths algorithm on the transformed network will find node-disjoint paths, but not necessarily achieve the desired result of a pair of disjoint paths with an even number of edges. With the addition

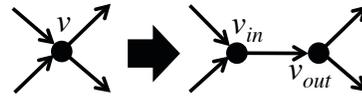


Fig. 3: Node splitting to find node-disjoint paths

of zero-cost edges to the transformed network, finding a pair of disjoint paths with an even number of edges in the transformed network may not yield paths with an even number of edges in the original network. A modification to the algorithm must be made to account for the new edges: in the transformed network, when choosing between an existing path from s to k , or some new path s to i plus a segment i to k , consider only segments that have an even number of “original” edges. This will ensure that a final path in the original network will have an even number of edges. The algorithm now begins to more closely resemble the Floyd-Warshall algorithm [22], which considers joining segments to find a shortest path. This new algorithm is called Even Shortest Pair of Node-Disjoint Paths.

These results can be extended to solve Wireless Guaranteed Protection problem with a single demand under 1-hop interference constraints. The maximum flow is found after every possible edge failure for each subgraph $G^e = (V, E \setminus e)$, $\forall e \in E$. The minimum of these maximum flows is the maximum protected flow. For each instance, we first see if there exists a pair of node-disjoint paths with an even total number of edges. If this exists, then a maximum flow of 1 is possible. If not, we check to see if there exists node-disjoint paths with an odd total number of edges (by running the standard edge-disjoint path routing algorithm on the transformed graph). If this exists, then a maximum flow of $\frac{2}{3}$ is possible. If no node-disjoint paths exist, then find some path from s to d , which can support a flow of $\frac{1}{2}$. Simulations show that the Even Shortest Pair of Node-Disjoint Paths algorithm is in fact optimal for all tested cases.

IV. AN OPTIMAL FORMULATION FOR WIRELESS GUARANTEED PROTECTION

In the previous section, an optimal solution for routing and scheduling with protection for a single demand was presented. While this provides insight, typical networks will need to simultaneously handle multiple connections. Additionally, many networks have interference constraints other than the 1-hop model. This section provides a mathematical formulation to the optimal solution for the Wireless Guaranteed Protection (WGP) problem with general interference constraints. In particular, for a set of demands, a route and schedule needs to be found such that after any link failure, all end-to-end connections maintain their same level of flow. For general interference constraints, the routing and scheduling problem was demonstrated to be NP-Hard [9]. We conjecture that adding protection constraints preserves NP-hardness; hence, a mixed integer linear program (MILP) is formulated to find an optimal solution to WGP.

In wired networks, a typical objective function for protection is to minimize the total allocated capacity needed to satisfy all demands. A similar objective cannot be clearly defined for

wireless networks since the concept of capacity changes in the presence of interference constraints. Consider some active link $\{i, j\}$. An adjacent link $\{j, k\}$ cannot be used simultaneously with $\{i, j\}$ because of interference; hence, simply adding additional link capacity (in a wired sense) will not allow its use. Another time slot must be allocated to allow a connection to use $\{j, k\}$ such that it does not interfere with $\{i, j\}$. Adding an additional time slot will reduce the time that each individual time slot in the schedule is active, which reduces the overall throughput of the network [8, 9, 13]. For example, consider a network with two time slots and a connection that supports a flow of 1 using these two time slots. If a third time slot is added to the schedule, then the original two time slots are only active for $\frac{2}{3}$ of the total time, and that flow's scheduled throughput is reduced from 1 to $\frac{2}{3}$. Thus, the objective we consider is to use a minimum number of time slots to route and schedule each demand with protection.

Finding a protection route and schedule using the minimum number of time slots allows for a simple comparison to existing wired and wireless protection schemes. The difference between the number of time slots necessary to route and schedule a set of demands before and after adding protection will be considered the reduction of the maximum throughput. To be consistent with the wireless protection scheme mentioned in [8], wireless flows are restricted to single paths (no flow splitting allowed). For ease of exposition, the MILP assigns the same throughput to all demands; see [19] for the formulation with different throughput requirements.

For the MILP, the following values are given:

- $G = (V, E)$ is the graph with a set of vertices and edges
- D is the set of flow requirements
- u_{ij} is the capacity of link $\{i, j\}$
- \mathcal{I} is the interference matrix, where $I_{ij}^{kl} \in \mathcal{I}$ is 1 if links $\{i, j\}$ and $\{k, l\}$ can be activated simultaneously, 0 otherwise
- \mathcal{T} is the set of time slots in the system, $\mathcal{T} \subset \mathbb{Z}^+$

The MILP solves for the following variables:

- x_{ij}^{sd} is a routing variable and is 1 if primary flow is assigned for demand (s, d) on link $\{i, j\}$, 0 otherwise
- $y_{ij,kl}^{sd}$ is a routing variable and is 1 if protection flow is assigned on link $\{i, j\}$ for the demand (s, d) after the failure of link $\{k, l\}$, 0 otherwise
- $\lambda_{ij}^{sd,t}$ is a scheduling variable and is 1 if link $\{i, j\}$ can be activated in time slot t for the demand (s, d) , 0 otherwise
- $\delta_{ij,kl}^{sd,t}$ is a scheduling variable and is 1 if link $\{i, j\}$ can be activated in time slot t after failure of link $\{k, l\}$ for the demand (s, d) , 0 otherwise
- s_t is 1 if time slot t is used by any demand, and 0 otherwise

The objective function is to minimize the number of time slots (the length of the schedule) needed to route all demands with protection:

$$\text{Objective: } \min \sum_{t \in \mathcal{T}} s_t \quad (2)$$

The following constraints are imposed to find a feasible routing and scheduling with protection.

Before a link failure:

- Flow conservation constraints for the primary flow: route primary traffic before a failure for each demand.

$$\sum_{\{i,j\} \in E} x_{ij}^{sd} - \sum_{\{j,i\} \in E} x_{ji}^{sd} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in V, \forall (s, d) \in D \quad (3)$$

- In any given time slot, for a given demand, only links that do not interfere with one another can be activated simultaneously.

$$\sum_{(s,d) \in D} \lambda_{ij}^{sd,t} + \sum_{(s,d) \in D} \lambda_{kl}^{sd,t} \leq 1 + I_{kl}^{ij}, \quad \forall \{i,j\} \in E, \forall \{k,l\} \in E, \forall t \in \mathcal{T} \quad (4)$$

- Only one demand can use a given link at a time.

$$\sum_{(s,d) \in D} \lambda_{ij}^{sd,t} \leq 1, \quad \forall \{i,j\} \in E, \forall t \in \mathcal{T} \quad (5)$$

- Ensure enough capacity exists to support the necessary flow for demand (s, d) on edge $\{i, j\}$ for the length of time that the link is active.

$$x_{ij}^{sd} \leq \sum_{t \in \mathcal{T}} \lambda_{ij}^{sd,t} u_{ij}, \quad \forall \{i,j\} \in E, \forall (s,d) \in D \quad (6)$$

- Mark if slot t is used to schedule a demand before a failure.

$$\lambda_{ij}^{sd,t} \leq s^t, \quad \forall \{i,j\} \in E, \forall t \in \mathcal{T}, \forall (s,d) \in D$$

After a link failure:

- Flow conservation constraints for protection flow: route protection traffic after each link failure $\{k, l\} \in E$.

$$\sum_{\{i,j\} \in E} y_{ij,kl}^{sd} - \sum_{\substack{\{j,i\} \in E \\ \{k,l\} \neq \{i,j\}}} y_{ji,kl}^{sd} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in V, \forall \{k,l\} \in E, \forall (s,d) \in D \quad (7)$$

- In any given time slot after the failure of link $\{k, l\}$, only links that do not interfere with one another can be activated simultaneously.

$$\sum_{(s,d) \in D} \delta_{ij,kl}^{sd,t} + \sum_{(s,d) \in D} \delta_{uv,kl}^{sd,t} \leq 1 + I_{uv}^{ij}, \quad \forall \{i,j\} \in E, \forall \{k,l\} \in E, \forall \{u,v\} \in E, \forall t \in \mathcal{T}, \forall \{i,j\} \neq \{k,l\} \neq \{u,v\} \quad (8)$$

- Only one demand can use a given link at a time after the failure of link $\{k, l\}$.

$$\sum_{(s,d) \in D} \delta_{ij,kl}^{sd,t} \leq 1, \quad \forall \{i,j\} \in E, \forall \{k,l\} \in E, \forall t \in \mathcal{T} \quad (9)$$

- Ensure enough capacity exists after the failure of link $\{k, l\}$ to support the necessary flow on edge $\{i, j\}$ for the length of time that the link is active.

$$y_{ij,kl}^{sd} \leq \sum_{t \in \mathcal{T}} \delta_{ij,kl}^{sd,t} u_{ij}, \quad \forall \{i,j\} \in E, \forall \{k,l\} \in E, \forall (s,d) \in D \quad (10)$$

- Mark if time slot t is used to schedule a demand after the failure of link $\{k, l\}$.

$$\delta_{ij,kl}^{sd,t} \leq s^t, \quad \forall \{i,j\} \in E, \forall \{k,l\} \in E, \forall t \in \mathcal{T}, \forall (s,d) \in D$$

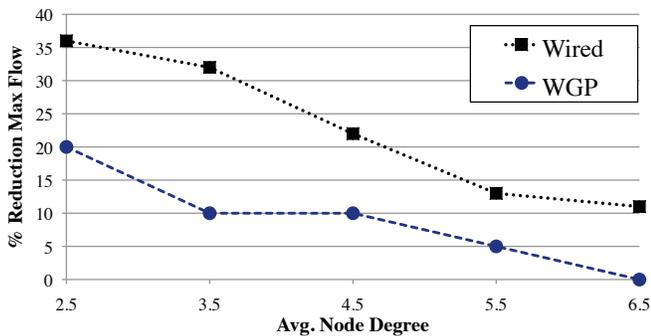


Fig. 4: Reduction of throughput when adding protection

To demonstrate how protection can be added to wireless networks with minimal reduction of throughput, WGP is compared to both the wired (without interference) and wireless protection (with interference) schemes. One hundred random graphs were generated with 25 nodes each. Nodes that are physically within a certain transmission range of one another are considered to have a link, and the transmission range is varied to give different desired average node degrees. The node degree is varied from 2.5 to 6.5, and for each graph, ten source/destination pairs are randomly chosen to be routed concurrently. All links have unit capacity; 1-hop interference constraints were used for the wireless networks. The simulation results are found in Fig. 4.

For comparison to wired protection, we use the same network topologies, however, in the wired case we do not enforce the interference constraints (i.e., all links can be activated simultaneously). For wired protection, we compute the reduction in throughput as the reduction in maximum flow after protection is added. As compared to the wired protection scheme, WGP has a lower reduction in throughput for all node degrees examined. For node degree 2.5, both the WGP and the wired protection schemes have larger reductions in throughput: 20% for WGP and 37% for wired. This is because at lower node degrees, there are fewer available end-to-end paths, and therefore after a failure, there are fewer routing options available. As the node degree increases, and there are more available end-to-end paths, the reduction in throughput decreases when adding protection. In fact, it is often possible for WGP to have no reduction in the throughput between the protected and unprotected setting. For an average node degree of 3.5, WGP only loses about 10% of throughput when adding protection, while the wired scheme loses 32%. For 20% of the simulations at node degree 3.5, there was no loss in throughput for WGP. When the node degree goes to 6.5, WGP no longer has any loss in flow, while the wired setting still has a loss of 11%.

We compare WGP to a wireless 1+1 protection scheme. In particular, wireless 1+1 protection applies the wired 1+1 protection scheme to wireless networks (as mentioned in [8]): i.e., find a schedule for the shortest pair of disjoint paths in the network between the source and destination, with the primary flow before a failure routed onto one path, and the backup flow routed onto the other. To compare WGP to wireless 1+1, the number of time slots needed beyond the non-protection routing are compared; these are the time slots needed to meet the

Avg. Node Degree	% Reduction of Protection Time Slots
2.5	72
3.5	63
4.5	60
5.5	52
6.5	46

TABLE I: WGP vs. Wireless 1+1

protection requirements. Table I shows the percent reduction in number of time slots needed to provide protection using WGP over wireless 1+1. When the average node degree is 2.5, WGP has up to a 72% reduction of time slots needed to meet protection requirements. The reason for this is that wireless 1+1 is scheduling two paths for each demand, a primary and a backup, and not trying to recapture any capacity after a failure; this in turn causes a significant increase in interference between connections. As the node degree increases, there is increased path diversity and more opportunities to find interference-free routings; hence, wireless 1+1 has better performance. But at all times, wireless 1+1 needs significantly more time slots to provide protection for all of the demands than WGP does, which is able to recapture capacity after a failure.

V. ALGORITHMS FOR PROVIDING WIRELESS PROTECTION

In the previous section, an MILP was presented to find an optimal solution to Wireless Guaranteed Protection (WGP), which is not a computationally efficient method of finding a solution. In this section, two time-efficient algorithms are presented to solve the Wireless Guaranteed Protection problem for a set of demands. Similar to the previous section, primary and backup flows are restricted to single paths, and the objective is to minimize the length of the schedule to route all demands with protection. We first show that this problem is NP-Hard under 1-hop interference constraints. Next, algorithms are developed assuming unit demands, unit capacity edges, and a single link failure model; the algorithms can be modified to reflect other values of demand and capacity. The algorithms are developed for dynamic (one-at-a-time) arrivals: an incoming demand needs to be routed and scheduled over an existing set of connections; the existing set cannot have their routings or schedules changed. A 1-hop interference model is used. We find that when compared to the optimal batch case (all connections are routed and scheduled simultaneously), the dynamic routing performs within a few percentage points of optimal.

First, in Section V-A, we demonstrate WGP to be NP-Hard under 1-hop interference constraints when flows are restricted to a single path. Next, in Section V-B, an algorithm to find a shortest 1-hop interference-free path using a minimal number of time slots is presented. This serves as the building block for the next two algorithms that are developed. In Section V-C, an algorithm for finding a minimal length schedule for WGP is presented, where a backup route and schedule is found for each possible failure. This approach has drawbacks in that after any failure, a new route is found; hence, a route and schedule for each failure event needs to be stored. To overcome this, an algorithm is developed in Section V-D using disjoint paths

such that only two paths are needed: a primary and a backup. In Section V-E, the performance of the two algorithms are compared to the optimal MILP formulation.

A. Complexity Results under 1-hop Interference Constraints

Without protection, the routing and scheduling problem is NP-Hard under general interference constraints [9]. But if flows for each demand are allowed to be split, a polynomial time algorithm is possible for 1-hop interference constraints [11]. We demonstrate that when flows cannot be split, the routing and scheduling problem becomes NP-Hard under 1-hop interference constraints.

Theorem 1. *Finding the minimum length schedule to route a set of demands under 1-hop interference constraints when flow splitting is not allowed is NP-Hard.*

We first consider the following necessary and sufficient condition for routing a set node-disjoint¹ pairs, (s_1, d_1) , ..., (s_N, d_N) , in only two time slots without flow splitting.

Lemma 4. *Under 1-hop interference constraints, a set of demands that are node-disjoint can be routed and scheduled using two time slots without flow splitting if and only if there exists node-disjoint paths between each of the node pairs.*

The proof is omitted for brevity, and can be found in [19]. Using Lemma 4, Theorem 1 can be quickly demonstrated.

Proof of Theorem 1: We reduce the Disjoint Connecting Paths Problem (DCPP) [23] to ours. DCPP asks the following question: given a graph $G = (V, E)$ and a collection of N node-disjoint pairs (s_1, d_1) , ..., (s_N, d_N) , does G contain N mutually node-disjoint paths, one connecting s_i and d_i for each i , $1 \leq i \leq N$? We can ask an equivalent question for our routing and scheduling problem: can a set of N node-disjoint pairs be routed and scheduled using the minimal number of time slots (two) under 1-hop interference constraints without flow splitting? If yes, then by Lemma 4 that means we have found N mutually node-disjoint paths, one connecting s_i and d_i for each i , $1 \leq i \leq N$, which solves DCPP. An answer of no means a solution to DCPP does not exist. ■

Next, we extend this complexity result to the case when protection is required.

Theorem 2. *Finding the minimum length schedule to route a set of demands with protection under 1-hop interference constraints without flow splitting is NP-Hard.*

The proof is omitted for brevity, but follows as an extension to the proof for Theorem 2; it can be found in [19].

B. Minimum Schedule for an Interference Free Path

We begin by developing an algorithm to find a shortest interference-free path using the minimum number of time slots under the 1-hop interference model. This algorithm will be a building block for the two protection algorithms that will be discussed in the upcoming sections. We consider an incoming demand for a connection between nodes s and d . Connections already exist in the network, with the set of \mathcal{T} time slots already

in use. Based on how the current connections are routed and scheduled, a set of edge interferences \mathbf{I} can be constructed, where for every edge $\{i, j\}$, $I_{ij} \in \mathbf{I}$ is the set of time slots that cannot be used on that edge because either that time slot is already used by $\{i, j\}$, or using that time slot on $\{i, j\}$ will interfere with another edge using it at that time. The set of edge interferences \mathbf{I} can be constructed in polynomial time, and will be given as an input to the algorithm.

First, we wish to determine the shortest interference free path without using any additional time slots beyond the set \mathcal{T} , and without rescheduling or rerouting existing connections. Each edge $\{i, j\}$ has a set of free time slots during which it can be used: $\tau_{ij} = \mathcal{T} \setminus I_{ij}$. Let P be the set of edges used in a path. If each edge of a loop-free path P has at least two free time slots, then that path can be scheduled without interference using the existing time slot allocation \mathcal{T} .

Lemma 5. *For 1-hop interference, a loop-free path P can be scheduled without interference if $|\tau_{ij}| \geq 2$, $\forall \{i, j\} \in P$.*

Proof: If $|\tau_{ij}| \geq 2$, $\forall \{i, j\} \in P$, then each edge in P has an available time slot that does not interfere with its adjacent set of edges. Since the path is loop-free, any two edges that use the same time slot will never be less than one hop apart from one another, and therefore never interfere with each other. ■

Using the result from Lemma 5, the following algorithm is constructed to find a 1-hop interference-free path using only the set of time slots \mathcal{T} : remove all edges in G that have $|\tau_{ij}| \leq 1$, find the shortest path P_{sd} between s and d , and assign time slots to the edges in P_{sd} such that it has an interference-free schedule.

An improvement can be made to the algorithm by attempting to maximize the number of free time slots on any edge, so that future connections will be less likely to require additional time slots to find an interference-free path. Currently, edges that have many free time slots are not given any preference. If an edge has only the minimal number of free time slots, it may be selected for use in a path. This may hurt finding interference-free paths for future connections by limiting the number of available time slots on an edge, thus necessitating new time slots. We assign a cost for each edge to be equal to the number of time slots that that edge interferes with: $c_{ij} = |I_{ij}|$. With respect to these new edge costs, a minimum-cost interference-free path is found. The more time slots an edge is in conflict with, the more expensive that edge will be, and the less likely it will be used in a route. We refer to this algorithm as `int_free_path`, which will return the edges and schedule of a path between s and d .

To find an interference free path that tries to minimize future conflicts, and using minimum additional time slots, we first find a minimum-cost interference free path for the current set of time slots assigned in the network, \mathcal{T} . If such a path does not exist, increase the set of available time slots by 1, and repeat. We note that the set of time slots will never increase by more than two since a feasible schedule can be found for any path with two free time slots. We call this algorithm `find_path`.

¹A node is a source or destination for at most one demand.

C. Minimum Length Schedule for Wireless Protection

In this section, an algorithm is developed that tries to find the minimum length schedule for the Wireless Guaranteed Protection problem, with an approach that is similar to the optimal solution found by the MILP in Section IV. The problem is broken up into $|E| + 1$ subproblems. First, the minimum length schedule is found to route the set of demands before a failure. Then, for each possible failure, the minimum length schedule is found to route the set of demands on a failure graph $G^{kl} = (V, E \setminus \{k, l\})$, $\forall \{k, l\} \in E$ (i.e., the graph that remains after the failure of edge $\{k, l\}$). Each of the solutions to these subproblems represents the route and schedule necessary to meet the protection requirements for the set of demands before and after link failure.

The *maximum* of any of these minimum length schedules will be the length of the schedule needed to add protection to set of demands in a wireless network. The algorithm is called `minimum_protect`; it will return the set of paths and schedules for each demand, indicating which path and schedule to use after any link failure.

D. Disjoint Path Wireless Guaranteed Protection

In Section V-C, an algorithm was described to find the minimum number of time slots to route and schedule a set of demands with protection. After any failure, a new route is found; hence, many possible routing configurations exist, and a route and schedule for each failure event needs to be saved. A more desirable approach may be to limit the number of paths needed to only two: a primary and a backup. Before continuing with the development of the algorithm, a complexity result is presented regarding using disjoint paths to provide protection in a wireless network with 1-hop interference constraints. For a set of time slots \mathcal{T} , simply determining if any solution exists to WGP using disjoint paths is NP-Complete.

Theorem 3. *For an incoming connection between s and d , using disjoint paths to provide protection in a wireless network with 1-hop interference constraints for the set of time slots \mathcal{T} is NP-Complete.*

A reduction is performed from the Dynamic Shared-Path Protected Lightpath-Provisioning (DSPLP) [2]. The proof is omitted for brevity, and can be found in [19].

Our approach for developing an algorithm to solve WGP using disjoint paths is similar to the wireless 1 + 1 protection scheme described earlier; however, we take advantage of the time slot reuse that is possible before and after a failure, as well as the opportunity to share protection resources between failure disjoint demands. If an edge in a primary path P uses time slot t , then for 1-hop interference, all edges adjacent to that edge also cannot use t . After the failure of an edge in the primary path, the time slots used to route that path are no longer needed (since they are not being used). The time slots on the edges of the primary path that did not fail now can be reused for protection; furthermore, the time slots on the edges that interfered with the failed primary path also become free to use for protection.

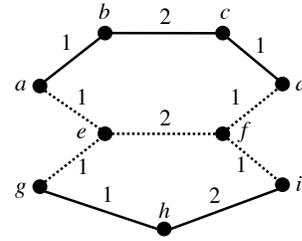


Fig. 5: Disjoint path routing and scheduling with protection

Protection resource sharing can also allow for time slot reuse. If two primary paths are failure disjoint under a single link failure model, only one will fail at a time. Hence, a time slot t on adjacent edges can be shared for protection between the two failure disjoint connections, since the two adjacent edges will never be activated simultaneously.

An example is shown in Fig. 5. Two demands need to be routed under 1-hop interference constraints: one from a to d , and another from g to i . Each edge is assigned a time slot, with the time slot labeling shown in the figure. The edges used for primary flow are indicated by solid lines, and the edges used for protection are dotted lines. After the failure of edge $\{a, b\}$, the entire primary path between a and d is no longer active, and its time slots will no longer be in use; hence, edges $\{a, e\}$ and $\{f, d\}$ can use time slot 1, even though they would have conflicted with $\{a, b\}$ and $\{c, d\}$ before the failure. Similarly, $\{g, e\}$ is assigned time slot 1, even though primary edge $\{g, h\}$ is assigned the same time slot. Since both primary paths are failure disjoint, time slot 2 on $\{e, f\}$ is shared between the two connections for protection. Additionally, because at most one backup path will be used at a time, protection edges $\{g, e\}$ and $\{a, e\}$ can both be assigned time slot 1; they will never interfere with one another. Similarly, $\{f, i\}$ and $\{f, d\}$ can be both assigned time slot 1.

This idea of time slot reuse after a failure forms the basis for the disjoint path wireless protection algorithm, which we label `disjoint_protect`. We consider an incoming demand requesting a connection between nodes s and d . Connections already exist in the network, with the set of \mathcal{T} time slots already in use. A interference-free primary path between s and d , P_{sd} , is found using `find_path`. Once a primary path fails, none of the time slots needed for that path, or on the edges that interfered with that path, are needed, and they become available to be used for protection. Next, a backup path B_{sd} is found that is disjoint to P_{sd} , and does not interfere with any of the other connections that did not fail. Additionally, the backup path B_{sd} will not interfere with the protection routings for the different existing demands that would fail if an edge in P_{sd} fails (i.e., B_{sd} will not interfere with the protection paths for demands whose primary paths are not disjoint with P_{sd}). Further details of the algorithm are omitted for brevity, and can be found in [19].

E. WGP Algorithm Simulations

The algorithms `minimum_protect` and `disjoint_protect` are compared to the optimal solution

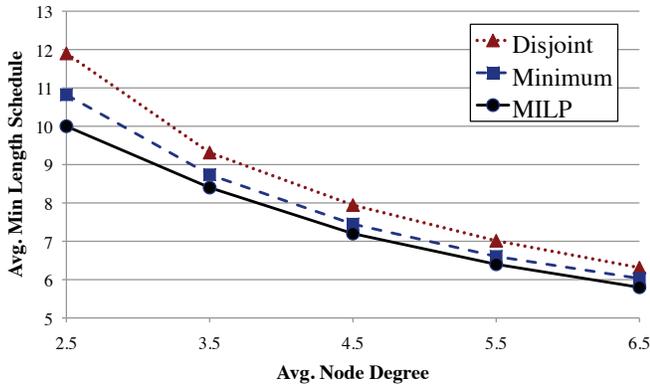


Fig. 6: Avg. time slots needed for WGP

found by the MILP in Section IV. A similar simulation setup is used as that in Section IV. One hundred random graphs were generated with 25 nodes each. The node degree is varied from 2.5 to 6.5, and for each random graph, ten source/destination pairs are randomly chosen to be routed concurrently, each with a unit demand. All links have unit capacity, and 1-hop interference constraints were used. The algorithms route and schedule demands one-at-a-time, while the MILP optimizes the route and schedule for all demands together (in batch). To compare the two, the algorithms randomly order the set of demands, and then solves for each demand one-at-a-time. The simulation results are found in Fig. 6.

Similar to the previous simulation, as node degree increased, the average minimum length schedule decreased. This is because of the increased diversity in possible number of end-to-end path, which leads to a greater opportunity of finding interference free paths. On average, minimum_protect needed only 4.5% more time slots to meet all requirements than the optimal MILP needed, and disjoint_protect needed 10.1% more time slots than the MILP.

VI. CONCLUSION

In this paper, the problem of guaranteed protection in a multi-hop wireless network is introduced. Because of link interference, resources that were unavailable prior to a failure can be used for protection after the failure. In fact, protection can often be provided using no additional resources. For the case of a single demand with 1-hop interference constraints, properties of an optimal solution are presented, and a polynomial time algorithm is developed that solves the problem of wireless routing and scheduling with and without protection. For general interference constraints and multiple concurrent demands, an optimal solution is developed for the protection problem via a mixed integer linear program. When compared to using traditional wired protection schemes on a wireless network, our Wireless Guaranteed Protection (WGP) scheme uses as much as 72% less protection resources to achieve the same level of resiliency. Two low-complexity algorithms to solve WGP are developed, and on average, these algorithms perform close to the optimal solution. A future direction for our work is to adapt the schemes developed in this paper to a distributed setting.

REFERENCES

- [1] S. Ramamurthy, L. Sahasrabudhe, and B. Mukherjee, "Survivable WDM Mesh Networks," *Journal of Lightwave Technology*, vol. 21, no. 4, p. 870, 2003.
- [2] C. Ou, J. Zhang, H. Zang, L. Sahasrabudhe, and B. Mukherjee, "New and Improved Approaches for Shared-Path Protection in WDM Mesh Networks," *Journal of Lightwave Technology*, vol. 22, no. 5, 2004.
- [3] B. Mukherjee, "WDM Optical Communication Networks: Progress and challenges," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1810–1824, 2000.
- [4] W. Yao and B. Ramamurthy, "Survivable Traffic Grooming with Path Protection at the Connection Level in WDM Mesh Networks," *Journal of Lightwave Technology*, vol. 23, no. 10, p. 2846, 2005.
- [5] A. Saleh and J. Simmons, "Evolution Toward the Next-Generation Core Optical Network," *Journal of Lightwave Technology*, vol. 24, no. 9, p. 3303, 2006.
- [6] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.
- [7] N. Ahmed, S. Kanhere, and S. Jha, "The holes problem in wireless sensor networks: a survey," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 4–18, 2005.
- [8] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem," in *Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM, 2003, pp. 42–54.
- [9] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless networks*, vol. 11, no. 4, pp. 471–487, 2005.
- [10] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proceedings of the 11th annual international conference on Mobile computing and networking*. ACM, 2005, pp. 58–72.
- [11] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *Information Theory, IEEE Transactions on*, vol. 34, no. 5, pp. 910–917, 1988.
- [12] W. Wang, Y. Wang, X. Li, W. Song, and O. Frieder, "Efficient interference-aware tdma link scheduling for static wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, 2006, pp. 262–273.
- [13] J. Zhang, H. Wu, Q. Zhang, and B. Li, "Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks," in *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*. IEEE, 2005, pp. 631–640.
- [14] R. Gupta, J. Musacchio, and J. Walrand, "Sufficient rate constraints for qos flows in ad-hoc networks," *Ad Hoc Networks*, vol. 5, no. 4, pp. 429–443, 2007.
- [15] X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE, 2007, pp. 1118–1126.
- [16] V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan, "Algorithmic aspects of capacity in wireless networks," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1. ACM, 2005, pp. 133–144.
- [17] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *Proceedings of the 12th annual int'l conf. on Mobile computing and networking*. ACM, 2006, pp. 227–238.
- [18] G. Sharma, C. Joo, N. Shroff, and R. Mazumdar, "Joint congestion control and distributed scheduling for throughput guarantees in wireless networks," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 21, no. 1, p. 5, 2010.
- [19] G. Kuperman and E. Modiano, "Providing Protection in Multi-Hop Wireless Networks," MIT, Tech. Rep., 2012. [Online]. Available: <http://web.mit.edu/gregk/www/papers/PPMHWN.pdf>
- [20] U. Derigs, "An efficient dijkstra-like labeling method for computing shortest odd/even paths," *Information processing letters*, vol. 21, no. 5, pp. 253–258, 1985.
- [21] J. Suurballe and R. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, 1984.
- [22] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, New Jersey, 1993.
- [23] M. Garey and D. Johnson, *Computers and intractability: A Guide to the Theory of NP-Completeness*. Freeman San Francisco, CA, 1979.