The Wireless Network Jamming Problem Subject to Protocol Interference

Author information blinded

December 22, 2014

Abstract

We study the following problem in wireless network security: Which jamming device placement configuration during a jamming attack results in the largest degradation of network throughput? Although others have studied similar jamming device placement problems, this paper is the first to include two important aspects: 1) network throughput is the optimization objective rather than network connectivity and 2) the network is subject to radio wave interference. We formulate this problem as a bi-level mixed-integer program, and solve it using a cutting plane approach that is able to solve networks with up to 81 transmitters. Experiments with the algorithm also yielded the following insights into wireless network jamming: 1) restricting the number of hops a message can travel did not significantly change the optimal throughput, 2) increasing the number of channels is the best strategy for designing a network that is robust against jamming attacks, and 3) increasing the range of the jamming devices is the best strategy for the attacker.

Keywords: cutting planes; network interdiction; security; wireless networks

1 Introduction

This paper addresses the problem of optimally placing a set of jamming devices within the boundary of a wireless communication network in order to minimize the throughput of the network during a jamming attack. This is the first study to include two important aspects of this problem: 1) throughput as the optimization objective and 2) the network is subject to radio interference. Thus, we call this problem the *jamming device location problem subject to interference* (JDLP-I). The goal of this study is to i) develop an optimization modeling approach that can incorporate these two previously unstudied aspects, ii) learn more about how to design a network to be robust against a jamming attack, iii) learn more about what actions a jammer might take to increase the disruption caused by a jamming attack.

1.1 Background and Motivation

The term wireless network is typically shorthand for wireless communication network and is a catch-all term for communication networks that require limited or no fixed infrastructure. A wireless

network consists of two or more transmitters, which communicate with each other by sending radio waves. These networks have long been used in the military, and in the last several decades they have become ubiquitous in everyday life. Because these networks require little or no infrastructure, they allow users almost unrestricted mobility. *Ad hoc* wireless networks (i.e., wireless networks that are implemented on an *ad hoc* basis) can be very useful in disaster response situations where much of the communication infrastructure is disabled.

Unfortunately, the characteristics that make wireless networks useful also make them vulnerable to attacks. The open-air medium has little physical defense against attacks. Not surprisingly, wireless network security has been an important issue for a considerable time, at least since World War II, and is increasing in prominence.

This two-fold truth–that wireless networks are very useful and inherently vulnerable–requires that we discover ways to minimize this vulnerability. The first step is to gain deeper understanding about why wireless networks are vulnerable and the extent of their vulnerability to certain types of attacks. This increased understanding will aid in the development of tools for designing and protecting wireless networks.

Responding to this need for understanding, many researchers have studied types of wireless network attacks and how to stop them. In addition, many have investigated strategic interactions between wireless network attackers and defenders. However, most of these studies have focused on single-hop networks, in which data transfer occurs only on a single link (for example, a wireless local area network (WLAN)). In these single-hop networks, it is usually easy to decide where to place the jamming devices. Thus, researchers often study strategic interactions between a fixed jamming and the operator of the network. Modeling the jamming of multi-hop networks is more difficult because an attacker has additional decisions regarding where where to place jamming devices. Although this problem has been studied by some, there is still much we do not know. Indeed, several researchers have highlighted this gap [33, 35].

1.2 Relevant Literature

The literature on the technological aspects of wireless communication security is abundant, and includes descriptions of jamming attacks [30, 18], anti-jamming strategies such as channel-hopping [28] and spread spectrum techniques [22], anti-jamming protocols for different network layers [3, 37], key management [10] and analysis of the effect of jamming on network performance measures [29, 4]. Other literature has examined operational considerations, such as the best method for responding to a jamming attack [24, 17] and developing efficient methods for determining the location of a jamming device [36].

While many studies have focused on strategic conflicts between network operators and jammers [20], and some studies have examined how a jammer would optimally allocate his or her resources [41], only a few studies have considered the actual placement of jamming devices [29, 8, 7].

However, the placement of devices to disrupt a network is the *sine qua non* of the field of network interdiction. After decades of advances, this field is well-established and we now have a

good understanding of how to interdict networks in order to achieve objectives such as minimizing the maximum flow [42], maximizing the shortest path [14], and minimizing network connectivity [2]. In addition, interdiction models have also been developed for networks with special structure such as hub-and-spoke[19] as well as trees and series-parallel networks [40]. Interdiction models have been applied to a wide variety of applications such as interdicting drug smuggling [42], interdicting a nuclear weapons project [6], interdicting nuclear smuggling Pan et al. [31], and analyzing the vulnerability of power grids [38].

Following this early work, researchers have studied variations of the canonical network interdiction problem, considering multiple commodities [21], multiple time periods [26], dynamic attacker/defender interactions [23], and several aspects of randomness such as a random interdiction effect [9], random network topology [12, 13], and random adversary characteristics [27, 32].

Indeed, the field of network interdiction has produced a mature set of modeling and algorithmic tools. However, these models and algorithms are tailored for *"wired"* networks, such as supply chains and road networks, and not for *wireless* networks. Thus, there is a need to extend this literature to the wireless domain.

Only a few studies lie near the intersection of wireless network jamming and network interdiction; namely, studies on the placement of jamming devices in order to minimize connectivity [29, 8, 7]. Connectivity is an important metric for wireless networks, especially when a jammer is able to disconnect the network. However, when the jammer does not have sufficient resources to disconnect the network, the throughput metric is probably more appropriate. In addition, none of the existing papers considers radio interference between transmitters. Thus, more work needs to be done on the problem of placing jamming devices; specifically, work that 1) considers throughput as an objective and 2) models radio interference between transmitters.

1.3 Contributions

The differences between wired and wireless networks preclude the simple extension of existing network interdiction models and algorithms to the wireless domain. For example, the throughput of a wired network can be computed in polynomial time, while for a wireless network under radio interference, this computation is NP-hard [16].

This paper studies the jamming device location problem subject to interference (JDLP-I). The main goals of research presented in this paper are to 1) develop a tractable approach for solving the JDLP-I and 2) increase our understanding of what makes jamming attacks more and less successful.

The results reported in this paper make the following contributions. 1) A mixed-integer programming (MIP) formulation, branch-and-cut procedure and Benders decomposition procedure for the JDLP-I; 2) empirical results that give insight into what actions are most effective for designing a network to be robust against jamming attacks and what actions are most effective for increasing the throughput reduction due to a jamming attack.

The remainder of this paper is as follows. Section 2 describes the JDLP-I, and Sections 3 and 4.1 describe a MIP formulation along with a branch-and-cut procedure. Section 5 contains the results

of an empirical study that examined 1) the tractability of the branch-and-cut procedure and 2) the relationship between various model parameters and network throughput during a jamming attack. These results provide insights into how to design a network that is robust against jamming, and how to increase the effectiveness of a jamming attack.

2 Problem Description

One way of describing JDLP-I is as a (two-level) Stackelberg game played in a 2-dimensional space. In this game, the jammer acts first and places omni-directional jamming devices among a candidate set of locations, $\mathcal{L}$, within the space. The cost of placing a jamming device at a location $\ell \in \mathcal{L}$ is r_ℓ, and the total cost of placing devices cannot exceed a budget R. Each jamming device has a transmission range e, beyond which signals from the device are too weak to cause any jamming. The objective of the jammer is to minimize the throughput of the network under the interference caused by the jamming devices. The jammer has access to an oracle that can compute the network throughput for any jamming device placement solution.

After the jammer acts, the network operator (likely a software program) routes and schedules traffic through a communication network. The network has the three layers. The *physical layer* consists of a set of omni-directional transmitting devices, each having a fixed location. Let a device be represented by a node, i, and let $\mathcal{N}$ be the set of all nodes. Each node has an infinite buffer for storing packets, as assumed in Jain et al. [16]. Let d_{ij} represent the Euclidean distance between nodes i and j. Each node has a communication range, c_i, within which it can communicate with another node, and an interference range, a_i, within which its transmissions can interfere with other signals. Both the communication range and interference range for a node i depend on the technology and power level of the transmitter located at node i. The communication rate between a pair of nodes i and j is u_{ij}.

The *connectivity layer* includes the transmission of packets between pairs of nodes. This layer is represented by a *connectivity graph*, denoted as $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{A}$ is the set of arcs, indexed by k. The connectivity graph is constructed by adding an arc between a pair of nodes i and j, both contained in the physical layer if node j is within the communication range of i, i.e., $d_{ij} \leq c_i$. Thus, node i can send to node j if arc (i, j) exists in G.

A jamming device disrupts the receipt of all messages sent to or received by nodes within its jamming range. Thus, an arc (i, j) is jammed by a device placed at location ℓ if $d_{\ell i} \leq e$ or $d_{\ell j} \leq e$.

In the absence of radio interference, the connectivity graph would be sufficient for computing a wireless network's throughput. However, because radio interference can have a significant effect on network throughput, wireless networks typically operate according to a communication protocol. The networks considered in this paper use the 802.11 medium access layer (MAC) protocol with virtual carrier sensing using the RTS-CTS exchange [16]. Under this protocol, a node i cannot send or receive at the same time as node j is sending or receiving if node j is within node i's interference range or node i is within j's interference range. This is because a successful communication between

a sender and receiver requires that the sender receives the link layer acknowledgment message sent by the receiver. Further, we use the *protocol model* of interference, rather than the physical one. Thus, in the model, arcs (i, j) and (p, q) interfere with each other if $d_{i'j'} \leq a_{i'}$ for $(i'j') = (i, q)$, (q, i), (i, p), (p, i), (j, p), (p, j), (j, q), (q, j).

The objective of the network operator is to maximize the interference- and jamming-affected throughput between a source node s and a destination node t, which is the rate of flow received by t. The operator routes and schedules flow with complete information about the location and power of the jamming devices, and has control over all flows in the network.

Thus, the two-level game can be represented as follows. Let $\mathbf{x}$ be a vector that indicates which jamming devices are located, and let X be the set of all feasible jamming location vectors. Further, let $\mathbf{y}$ be a flow in the network, $Y(\mathbf{x})$ be the set of allowable flows given jamming vector $\mathbf{x}$, and let $TH(\mathbf{y})$ be the network throughput for a given flow. The two-stage game can be formulated as the following bi-level optimization problem:

$$\min_{\mathbf{x} \in X} \max_{\mathbf{y} \in Y(\mathbf{x})} TH(\mathbf{y})$$

3 Bi-Level Mixed-Integer Programming Model Formulation

We capture the interference between arcs by modeling a third network layer using a *conflict graph*, first proposed by Jain et al. [16]. The conflict graph, denoted as G', has a node for each arc in the connectivity graph; thus, a node in G' is denoted as (i, j). An arc exists between nodes (i, j) and (p, q) in G' if arcs (i, j) and (p, q) in G interfere with each other.

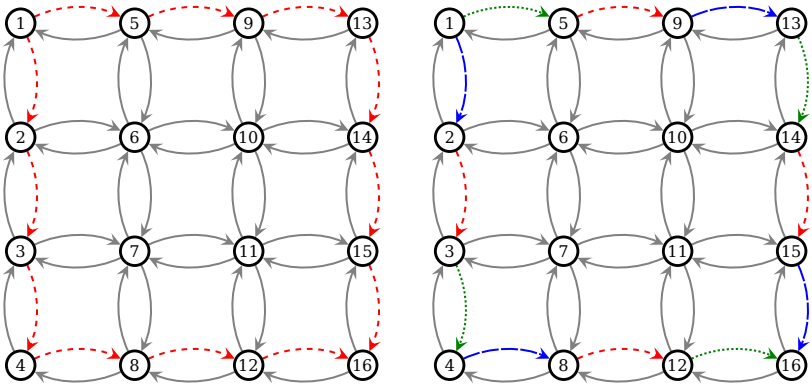
Because arcs interfere with each other, they can not all transmit data simultaneously. Thus, the network must schedule data transmission in such a way that arcs alternate between the active and inactive states in order to maximize throughput while avoiding data packet collisions caused by arc interference. Jain et al. [16] showed that sets of active arcs will not interfere with each other if the arcs in each set form an independent set in the conflict graph, G'. It is not difficult to show that a maximal throughput is achieved by only using *maximal* independent sets as active arc sets, where a maximal independent set is one in which adding a node to the set causes it to no longer be independent. Let $\mathcal{I}$ represent the set of all maximal independent sets, $\{I_1, I_2, \ldots, I_K\}$ as well as the set of their indices $\{1, 2, \ldots, K\}$. Further, let $\mathcal{I}_k$ be the set of all maximal independent sets that contain arc k.

Given this notation, the steady-state scheduling and routing of data in the network can be described using the following two vectors of variables. First, let $\mathbf{w} = (w_n)_{n \in \mathcal{I}}$ be a *usage vector*, whose elements define the fraction of time that all of the arcs in independent set I_n are simultaneously active. Second, let $\mathbf{y} = (y_k)_{k \in \mathcal{A}}$ be a a vector whose elements y_k represent the average flow on arc k; these variables represent the average flow because data transmission occurs by alternating between different active arc sets, meaning that each arc is only active for a fraction of the time. Thus, the value of y_k is constrained by the average capacity of k, which is the product of the capacity of k,

u_k and the total proportion of time that k is active, $\sum_{n \in \mathcal{I}_k} w_n$.

Figure 1 illustrates the difference between a maximum flow in a wired network versus a wireless one. (All figures in this article were produced using the `graph-tool` Python package [34].) The network shown is a 4x4 grid network with a single source (1) and sink (16) and each link having unit capacity. Figure 1a shows a maximum flow of 2.0 in the wired network, which has no interference. Unit flows are is sent on routes $1-2-3-4-8-12-16$ and $1-5-9-13-14-15-16$, represented by the red dashed lines. In this solution, arcs are always active because the network is not subject to interference.

Figure 1b shows the maximum flow in the wireless network, which is subject to arc interference (each node has interference range 1.0). Again, the flow is sent on the same two routes. However, because of interference, three alternating sets of active arcs are used 1/3 of the time: $\{(1,5),(3,4),(13,14),(12,16)\}$ `(green dotted)`, $\{(2,3),(5,9),(8,12),(14,15)\}$ `(red dashed)`, and $\{(1,2),(4,8),(9,13),(15,16)\}$ `(blue long dashed)`. Thus, each arc on the two paths is active only 1/3 of the time and the flow is now 2/3.



(a) Wired network = 2 (b) Wireless network = 2/3

Figure 1: Wired vs. wireless network throughput

We now explain the theoretical foundation for the solution in Figure 1.

Definition 1. A $n \times n$ *unit grid network* is a $n \times n$ grid network in which each row is one unit from its neighbor and each column is one unit from its neighbor. Further, the communication and interference range for each node is 1.

Lemma 1. *The average flow of an arc on a path of length two or three in a unit grid network is no more than $u/2$ and $u/3$, respectively.*

Proof.

Case 1: path of length 2. Let the path be defined by nodes i, j, and k. By the definition of a unit grid network, $d_{i'j'} = a_{j'}$ for all (i', j') in $\{(i,j), (j,i), (j,k), (k,j)\}$. Thus, each of the two arcs interferes with each other and in a maximum flow solution each of the arcs must be in a separate independent set. Since only one independent set can be active at one time in a maximum flow solution, the proportion of time that either of the two independent sets is active is $1/2$. When active, each arc can transmit at most u units of flow. Thus, the average flow of any arc is at most $u/2$.

Case 2: path of length 3. (Following the same argument as in Case 1.) Let the path be defined by nodes i, j, k, and ℓ. By the definition of a unit grid network, $d_{i'j'} = a_{j'}$ for all (i', j') in $\{(i,j), (j,i), (j,k), (k,j), (k,\ell), (\ell,k)\}$. Thus, each of the three arcs interferes with each other. Thus, in a maximum flow solution, each of the arcs must be in a separate independent set. Since only one independent set can be active at one time in a maximum flow solution, the proportion of time that either of the three independent sets is active is $1/3$. When active, each arc can transmit at most u units of flow. Thus, the average flow of any arc is at most $u/3$. □

Lemma 2. *The amount of flow on a path in a unit grid network is at most $u/2$ if the path has length 2 and $u/3$ if it has length of at least 3.*

Proof. Consider a path of length two. Since both arcs on the path are part of a path of length two, their maximum flow is $u/2$ (Lemma 1) and the maximum flow of the path is at most $u/2$. Now, consider a path of length three. By Lemma 1, the maximum flow on any arc in the path is $u/3$ so the flow on the path is at most $u/3$. □

Proposition 1. *Consider a $n \times n$ unit grid network from a single source located on a corner and a single sink located on a corner. Let u be the capacity of every arc. The throughput is at most u if $n = 2$ and at most $2u/3$ if $n > 2$.*

Proof. When $n = 2$, every arc in the network interferes with every other arc. Consequently, only one arc can transmit at one time. In one optimal solution, each of four arcs is in its own independent set (another is to use only one path with two arcs and have two independent sets). In this optimal solution, each arc is active $1/4$ of the time, so the the throughput is u.

When $n > 2$, at most two arcs can send from from the sink since it is a corner node. Since $n > 2$, each of these arcs must be part of a path of length 3. Consequently, these arcs have an average flow of at most $u/3$ (Lemma 2). Thus, the throughput is constrained by the sum of the upper limits on the two arcs, which is $2u/3$. □

Remark 1. A well-known property of the maximum flow problem is that if the arc capacities are integer, then an integral maximum flow solution exists [1]. In a $n \times n$ unit grid network with $n > 2$, this is only true if the arc capacities u are a multiple of 3. (To show this, notice that the proof of Proposition 1 shows that the links coming out of the source node each have an average flow of at most $u/3$.)

3.1 Bi-level Formulation

Let $\mathbf{x} = (x_\ell)_{\ell \in \mathcal{L}}$ be a vector of binary variables, with $x_\ell = 1$ if a jamming device is placed at location ℓ and 0 otherwise. Further, let y_a be the throughput of the network, i.e., rate of flow into the sink t. The bi-level mixed-integer formulation is as follows.

$$\min_{\mathbf{x} \in \{0,1\}^{|\mathcal{L}|}} \quad g(\mathbf{x}) \tag{1a}$$

$$\text{s.t.} \qquad \sum_{\ell \in \mathcal{L}} r_\ell x_\ell \leq R, \tag{1b}$$

where

$$g(\mathbf{x}) = \quad \max \quad y_a \tag{1c}$$

$$\text{s.t.} \quad \sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k = \begin{cases} y_a & i = s \\ 0 & i \in \mathcal{N} \setminus \{s, t\} \\ -y_a & i = t \end{cases} \quad \forall i \in \mathcal{N} \quad [\alpha_i], \tag{1d}$$

$$0 \leq y_k \leq \left(\sum_{n \in \mathcal{I}_k} w_n \right) u_k, \quad \forall k \in \mathcal{A} \quad [\beta_k], \tag{1e}$$

$$\sum_{n \in \mathcal{I}} w_n \leq 1 \quad [\gamma], \tag{1f}$$

$$w_n \geq 0 \quad \forall n \in \mathcal{I} \tag{1g}$$

$$0 \leq y_k \leq u_k(1 - x_\ell), \quad \forall k \in \mathcal{A}, \ell \in \mathcal{L}_k. \tag{1h}$$

The inner problem, $g(\mathbf{x})$, computes the network throughput given a jamming placement solution. (Jain et al. [16] showed that the maximum flow problem, augmented with independent set usage variables (w_n) to account for interference, can be used to compute the throughput of a network subject to interference.) The outer level objective (1a) is to minimize the network throughput by strategically placing jamming devices. The inner level objective (1c) is to maximize the throughput. Constraints (1d) balance flow at each intermediate node and require that the source sends flow equal to the network throughput, while the sink receives flow equal to the network throughput. Constraints (1e) constrain the average flow of each arc to be non-negative and no more than the proportion of time the arc is active multiplied by the arc's capacity. Constraints (1f) and (1g) require the usage vector to be a probability. Finally, letting $\mathcal{L}_k$ be the set of locations within jamming range of arc k, the last constraint (1h) sets the capacity of a jammed arc to zero. (The dual variables for selected constraints are denoted by the variables in brackets.)

3.2 Cormican reformulation

A standard method of solving formulation (1) is to fix the outer problem variables ($\mathbf{x}$), take the dual of the inner maximization problem, and then unfix the outer problem variables and solve the resulting minimization problem using well-known methods for solving mixed-integer programs. However, a well-known drawback of formulation (1) is that taking the dual of the inner maximization problem leads to a resulting minimization problem with bi-linear terms due to constraint (1h). Thus, Cormican et al. [9] reformulated the inner maximization problem by penalizing flow on jammed arcs and eliminating the constraint (1h), resulting in a reformulation equivalent to (1). Their reformulated inner problem is as follows:

$$\begin{aligned} g(\mathbf{x}) = \quad & \max_{\mathbf{y}\geq\mathbf{0}} \quad y_a - \sum_{k\in\mathcal{A}}\sum_{\ell\in\mathcal{L}_k} x_\ell y_k && \text{(2a)} \\ & \text{s.t.} \qquad \text{(1d)–(1g).} \end{aligned}$$

Now we take the dual of the inner maximization problem and obtain the following minimization problem.

$$\begin{aligned} \min_{\mathbf{x}\in X,\alpha,\beta,\gamma} \quad & \gamma && \text{(3a)} \\ \text{s.t.} \quad & \alpha_i - \alpha_j + \beta_k + \sum_{\ell\in\mathcal{L}_k} x_\ell \geq 0 \quad \forall (i,j) = k \in \mathcal{A}, && \text{(3b)} \\ & \alpha_t - \alpha_s \geq 1, && \text{(3c)} \\ & \gamma \geq \sum_{k\in I_n} u_k \beta_k \quad \forall n \in \mathcal{I}, && \text{(3d)} \\ & \alpha_i \text{ unrestricted} \quad \forall i \in \mathcal{N}, && \text{(3e)} \\ & \beta_k \geq 0 \quad \forall k \in \mathcal{A}, && \text{(3f)} \end{aligned}$$

where $X = \{x_\ell : \ell \in \mathcal{L}, x_\ell \in \{0,1\}, \sum_{\ell\in\mathcal{L}} r_\ell x_\ell \leq R\}$ is the set of feasible jamming location solutions.

Remark 2. For the maximum flow network interdiction problem (a special case of our problem in which there is no interference), the integrality property of the maximum flow problem is used to show that an optimal solution exists in which $\alpha_i \in \{0,1\}$ for all $i \in \mathcal{N}$ and $\beta_k \in \{0,1\}$ for all $k \in \mathcal{A}$. However, this is not always the case for formulation (3), as shown in the following example. This is not surprising because the underlying maximum flow variation does not have the integrality property in all cases (see Remark 1).

Example 1. Consider the 3x3 unit grid network shown in Figure 2. The jamming range is set to 0.0. With one jamming device allowed to be placed (i.e., $r_\ell = 1$, $R = 1$), the throughput with formulation 3 is 0.4. However, when the α_i are restricted to be integral, the throughput is 1.0.

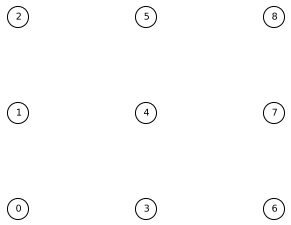


Figure 2: Network for Example 1

An immediate drawback of formulation (3) is the exponential number of constraints (3d), which we will address in Section 4.1.

3.3 Straightforward Extensions

Following the work of Jain et al. [16], who provided extensions for their network throughput model, there are several straightforward extensions of the JDLP-I:

3.3.1 Multiple Communication Pairs

Rather than having a single source and sink, the model is easily extended for multiple sources and sinks. Let $\mathcal{M}$ be a set of source-sink pairs $\{(s_1, t_1),(s_2, t_2),\ldots,(s_M, t_M)\}$ and the set of their indices be $\{1, 2, \ldots, M\}$. Each pair has a desired communication flow rate between source and sink, D_m, i.e., the sinks are not greedy [16]. Replace the flow variables y_k with y_{km}, which denotes the flow from s_m to t_m on arc k and modify the objective function and the constraints involving y_k accordingly. Further, the following demand constraint must be added to formulation (1):

$$\sum_{k \in \mathcal{A}} y_{km} \leq D_m \quad \forall m \in \mathcal{M} \quad [\epsilon_m].$$

3.3.2 Latency Constraints

To prevent excess latency, one might wish to limit the length of a flow path. Let $\mathcal{P}$ be the set of all $s - t$ paths with a length less than a defined value. The following path-based formulation can be used to compute the network throughput given a jamming solution $\mathbf{x}$ and a constraint on path length.

$$
\begin{aligned}
g(\mathbf{x}) = \quad & \max \quad \sum_{p \in \mathcal{P}} \left(y_p - \sum_{\ell \in \mathcal{L}_p} x_\ell y_p \right) && \text{(4a)} \\
& \text{s.t.} \quad 0 \leq \sum_{p \in \mathcal{P}_k} y_p \leq \left(\sum_{n \in \mathcal{I}_k} w_n \right) u_k, \quad \forall k \in \mathcal{A}, && \text{(4b)} \\
& \quad (1f)\text{–}(1g) && \text{(4c)} \\
& \quad y_p \geq 0 \quad \forall p \in \mathcal{P}. && \text{(4d)}
\end{aligned}
$$

Thus, the single-level formulation is

$$
\begin{aligned}
& \min_{\mathbf{x} \in X, \beta, \gamma} \quad \gamma && \text{(5a)} \\
& \text{s.t.} \quad \sum_{k \in \mathcal{A}_p} \beta_k + \sum_{\ell \in \mathcal{L}_p} x_\ell \geq 1, \quad [y_p] \quad \forall p \in \mathcal{P}, && \text{(5b)} \\
& \quad (3d)\text{–}(3f).
\end{aligned}
$$

Path length can be constrained by defining the set $\mathcal{P}$ to only contain paths less than a certain length.

3.3.3 Multiple Channels

To include multiple communication channels, one only needs to include multiple arcs between every pair of nodes in the connectivity graph. Arcs that are of the same channel do not interfere with each other.

3.3.4 Other

Jain et al. [16] also mention several other extensions to their model for the maximum flow problem subject to interference, which we did not attempt to extend to the jamming case. These include multiple radios per node, directional antennas, multirate radios, other models of interference, and equity-based objective functions. We believe that making these extensions to our model would not be difficult.

4 Solution Methodology

We investigated two methods for solving formulation (3): branch-and-cut and Benders decomposition.

4.1 Branch-and-Cut

We account for the large number of usage variables in formulation (3), by using a cutting plane approach that generates them dynamically. Let $\bar{\mathcal{I}}$ be a subset of $\mathcal{I}$. The restricted version of formulation (3) is then the following:

$$\begin{aligned} \min_{\mathbf{x}\in X,\alpha,\beta,\gamma} \quad & \gamma && \text{(6a)}\\ \text{s.t.} \quad & \text{(3b),(3c),(3e),(3f),} && \text{(6b)}\\ & \gamma \geq \sum_{k\in I_n} u_k\beta_k \quad \forall n \in \bar{\mathcal{I}}. && \text{(6c)} \end{aligned}$$

Rather than solving formulation (3) in its entirety, we solve formulation (6) and add additional independent sets to $\bar{\mathcal{I}}$ in an *as-needed* fashion, i.e., via a cutting-plane approach. Using this approach, we seek to iteratively construct a set $\bar{\mathcal{I}}$ in a way that meets the following criteria: 1) the size of $\bar{\mathcal{I}}$ is much smaller than the size of $\mathcal{I}$ and 2) $z^*(\bar{\mathcal{I}}) = z^*(\mathcal{I})$, where $z^*(\mathcal{I})$ is the optimal objective value obtained when the model is limited to the set of maximal independent sets $\mathcal{I}$. Using this construction, we may avoid much of the computation time needed to compute $z^*(\mathcal{I})$ by solving a sequence of much smaller problems, i.e., $z^*(\bar{\mathcal{I}}_1), z^*(\bar{\mathcal{I}}_2), \ldots, z^*(\bar{\mathcal{I}}_{K'})$, where $\bar{\mathcal{I}}_1 \subset \bar{\mathcal{I}}_2 \subset \bar{\mathcal{I}}_{K'}$ and K' is the number of iterations needed for the cutting plane algorithm to return an optimal solution.

However, to generate new constraints of type (6c), which corresponds to generating new independent sets, we solve a separation problem that seeks to find an independent set whose addition to $\bar{\mathcal{I}}$ causes constraints (6c) to be maximally violated given the current fixed values of γ and $\boldsymbol{\beta}$. Let $\mathcal{N}(G')$ be the set of nodes in the conflict graph. Let $\mathcal{A}(G')$ be the set of arcs in the conflict graph. The separation problem is as follows:

$$\begin{aligned} z(\boldsymbol{\beta}) = \quad \max \quad & \sum_{k\in\mathcal{N}(G')} \hat{\beta}_k u_k v_k && \text{(7a)}\\ \text{s.t.} \quad & v_k + v_{k'} \leq 1 \quad \forall (k,k') \in \mathcal{A}(G'), && \text{(7b)}\\ & v_k \in \{0,1\} \quad \forall k \in \mathcal{N}(G'). && \text{(7c)} \end{aligned}$$

The objective (7a) seeks to maximize the righthand side of constraints (6c). Constraints (7b) require the set defined by the v_k variables to be an independent set (see Section 3).

We embed the cutting plane procedure inside a branch-and-bound algorithm (B&B). That is, branch-and-bound is applied to the restricted master problem (6). Whenever B&B identifies a new incumbent solution, we execute a *separation procedure* to potentially add a new cutting plane. Let $\hat{\gamma}$ and $\hat{\boldsymbol{\beta}}$ be the incumbent values of γ and $\boldsymbol{\beta}$, respectively, at the point in which a new incumbent solution is found for $\mathbf{x}$. The separation procedure is as follows:

1. Solve the separation problem (7) given $\hat{\boldsymbol{\beta}}$, returning optimal solution $\mathbf{v}^*$.

2. If $\hat{\gamma} < \sum_{k \in \mathcal{N}(G')} u_k \hat{\boldsymbol{\beta}}_k v_k^*$, then append the independent set $\{k : k \in \mathcal{N}(G'), v_k^* = 1\}$ to $\bar{\mathcal{I}}$.

4.2 Accelerated Benders Decomposition

4.2.1 Benders Decomposition

The standard Benders decomposition approach begins with reformulating the single-level formulation (3) as the following two-level formulation:

$$\min_{\mathbf{x} \in X} \quad g(\mathbf{x}) \tag{8}$$

where

$$\begin{aligned}
g(\mathbf{x}) = \quad & \min_{0 \le \boldsymbol{\beta} \le 1, \gamma \ge 0} \quad \gamma && && \text{(9a)}\\
& \text{s.t.} && \alpha_i - \alpha_j + \beta_k + \sum_{\ell \in \mathcal{L}_k} \hat{x}_\ell \ge 0 \quad \forall (i,j) = k \in \mathcal{A}, \quad [y_k] && \text{(9b)}\\
& && \alpha_t - \alpha_s \ge 1, \quad [y_a] && \text{(9c)}\\
& && \gamma - \sum_{k \in I_n} u_k \beta_k \ge 0 \quad \forall n \in \mathcal{I}, \quad [w_n] && \text{(9d)}\\
& && \alpha_i \text{ unrestricted} \quad \forall i \in \mathcal{N}, && \\
& && \beta_k \ge 0 \quad \forall k \in \mathcal{A}. && \text{(9e)}
\end{aligned}$$

The *Benders master problem* is then

$$\begin{aligned}
& \min_{\theta \ge 0, \mathbf{x} \in X} \quad \theta && && \text{(10a)}\\
& \text{s.t.} && \theta \ge y_a^\iota - \sum_{k \in \mathcal{A}} \sum_{\ell \in \mathcal{L}_k} y_k^\iota x_\ell \quad \forall \iota = 1, \ldots, \mathbb{I}, && \text{(10b)}
\end{aligned}$$

where $\mathbb{I}$ is the number of extreme point solutions to the dual of (9). Constraints (10b) define supporting hyperplanes of the function $g(\mathbf{x})$ and are known as *Benders optimality cuts*. Because the cardinality of $\mathbb{I}$ is usually very large, the Bender's decomposition algorithm dynamically generates these optimality cuts, computing them using the *Benders subproblem*, which is the dual of (9). An advantage of using Benders decomposition to solve bi-level min-max problems such as (8) is that the Benders subproblem is precisely the follower's maximum flow problem (2).

However, solving the follower's maximum flow problem (2) is challenging because of the large number of w variables, each corresponding to independent sets. Thus, we employ a simple column generation procedure. The *column generation master problem* is the following linear program

$$
\begin{aligned}
g(\mathbf{x}) \quad \max_{\mathbf{y}\geq\mathbf{0}} \quad & y_a - \sum_{k\in\mathcal{A}} \sum_{\ell\in\mathcal{L}_k} x_\ell y_k && \text{(11a)}\\
\text{s.t.} \quad & \text{(1d)},\\
& 0 \leq y_k \leq \left(\sum_{n\in\bar{\mathcal{I}}_k} w_n \right) u_k, \quad \forall k \in \mathcal{A} \quad [\beta_k], && \text{(11b)}\\
& \sum_{n\in\bar{\mathcal{I}}} w_n \leq 1 \quad [\gamma], && \text{(11c)}\\
& w_n \geq 0 \quad \forall n \in \bar{\mathcal{I}}, && \text{(11d)}
\end{aligned}
$$

where $\bar{\mathcal{I}}$ is the *restricted* set of maximum weight independent sets. We generate new columns (i.e., maximum weight independent sets), by using the maximum weight independent set problem (7) as a pricing problem. Algorithm 1 describes the column generation procedure.

Algorithm 1 Column generation procedure for throughput problem.

function ComputeThroughputColumnGeneration($\mathbf{x}$)
  initialize. Set $\gamma^* = 0$ and $z(\boldsymbol{\beta}^*) = \infty$.
  while $\gamma^* < z(\boldsymbol{\beta})^*$ **do**
    Solve the master problem (11), returning, $g(\mathbf{x})^*$, $\mathbf{y}^*$, $\boldsymbol{\beta}^*$, and γ^*.
    Solve the pricing problem (7), returning $z(\boldsymbol{\beta}^*)$ and independent set I^*.
    if $\gamma^* < z(\boldsymbol{\beta})^*$ **then** add new w variable corresponding to I^* to master problem.
  return The optimal flow $\mathbf{y}^*$ and throughput $g(\mathbf{x})^*$.

The Benders decomposition procedure starts with an initial value of $\mathbf{x}$ and a lower bound on θ and then iteratively adds new Benders optimality cuts via the Benders subproblem. Thus, after iteration $\bar{\mathbb{I}}$ the Benders master problem has exactly $\bar{\mathbb{I}}$ optimality cuts. (Note that the Benders subproblem has relatively complete recourse, which means that for all $\mathbf{x} \in X$ there exists a feasible solution to the Benders subproblem; thus, Benders feasibility cuts are not needed.) Algorithm 2 defines the Benders decomposition algorithm.

Algorithm 2 Benders decomposition procedure.

function BendersDecomposition
  initialize. Set $\mathbf{x} \leftarrow \mathbf{0}$, $lb \leftarrow 0$ and $ub \leftarrow \infty$.
  while $lb < ub$ **do**
    Solve the master problem 10, returning θ^* and $\mathbf{x}^*$.
    Set $lb \leftarrow \theta^*$.
    Solve subproblem (2), returning optimal flow $\mathbf{y}^*$ and throughput $g(\mathbf{x}^*)$.
    Add Benders optimality cut $\theta \geq y_a^* - \sum_{k\in\mathcal{A}} \sum_{\ell\in\mathcal{L}_k} y_k^* x_\ell$ to the Benders master problem.
    if $g(\mathbf{x}^*) < ub$ **then** set $ub \leftarrow g(\mathbf{x}^*)$ and $\mathbf{x} \leftarrow \mathbf{x}^*$.
  return The optimal locations $\mathbf{x}$ and throughput ub.

5 Computational Results

In this section, we use our optimization model to provide insight into several questions.

1. *Will constraining the number of hops allowed on a path decrease the optimal throughput?* In some wireless networking applications, packet latency is a concern. Thus, some benefit could be gained in ensuring low latency by constraining the number of hops on a path.

2. *How long does it take to solve the model and what size instances can be solved?*

3. *What strategies for designing a network to be jamming-resistant are most beneficial?* We investigate the effect of changing the density of nodes, communication range, interference range, and number of communication channels. We also compare two types of networks to provide insight on the effect of arranging transmitters differently.

4. *What strategies for maximizing the impact of a jamming attack are most beneficial?* We investigate the effect of changing the number of potential jamming device locations, the jammer's budget, and the jamming range.

5.1 Experimental Setup

All experiments were performed using the cutting plane algorithm described in Section 4.1 applied to the Cormican formulation (3). Two extensions were added to the model: multiple communication pairs (see Section 3.3.1) and multiple channels (see Section 3.3.3).

The experiments were run on two types of networks. The first is a $n \times n$ grid network; Figure 2 shows a 3×3 grid network. The second type is a 54-node network assembled at Carnegie Mellon University's Intel Berkeley Research Lab [5], which we denote as "CMU"; the network is shown in Figure 3. The default grid network is a 7×7 grid network, which has a node count (49) comparable to the number of nodes in the CMU network. For all networks, the nodes were placed inside a unit square. Thus, the lateral and vertical distance between nodes in the grid network is $\frac{1}{N-1}$. In the default instance, each network has a single channel.

For each of the networks, 16 communication pairs were used, each having a desired communication rate generated randomly from the interval $[0, 2]$. For the grid networks, the orgin and destinations were placed at the corners of the network and at the midpoints of the edges. For the CMU dataset, the origin and destination of each communication pair were generated randomly without replacement from the set of nodes. The default communication range used was 1/6, which is the lateral and vertical distance between nodes in the default 7×7 grid network. The interference range was set to $a_i = 1.75c_i$ for all $i \in \mathcal{N}$; the multiplier 1.75 was recommended by Iyer et al. [15]. The set of possible jamming devices locations, $\mathcal{L}$, is an NxN grid overlaid on top of the communication network. The default is a 5×5 grid, resulting in 25 possible locations. The jamming range e, was set to 1/3 for all jamming devices.

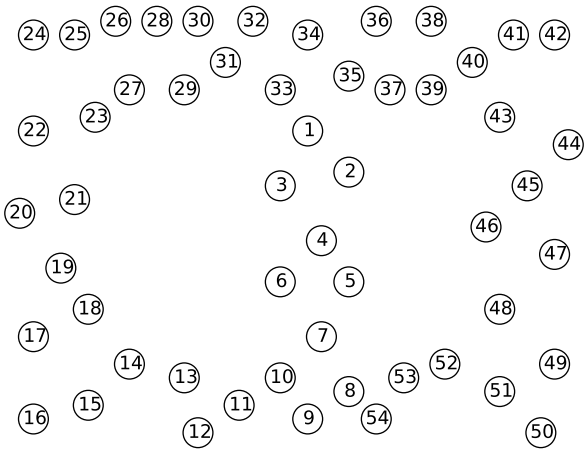


Figure 3: CMU network

The following model parameter values were used in all of the experiments. The cost of locating a jamming device, r_ℓ, was set to 1 in all experiments, so the jammer's budget, R, was set to integer values. The arc capacity was set to $u_k = 1$ for all arcs $k \in \mathcal{A}$.

Table 1 lists the default parameter values used in all the experiments, unless otherwise indicated.

Table 1: Dataset parameter values used in experiments

Parameter	**Baseline value**
Network dataset	grid_7x7
Number of channels	1
Number of communication pairs	16
Communication rate for each pair, D_m	$Unif(0, 2)$
Communication range, c_i	$\frac{1}{6}$
Interference range multiplier	1.75 [15]
Number of possible jamming devices locations, $\lvert\mathcal{L}\rvert$	25
Jamming range, e	1/3
Number of jamming devices, R	2

5.2 Effect of Limiting the Number of Hops

We used the path-based formulation (5) to vary the number of hops allowed on a path and we observed the effect on throughput. Rather than including all paths in the formulation, we generated them dynamically using a delayed row generation approach, which dynamically adds Constraints

(5b). Specifically, the separation problem is a shortest path problem on the connectivity graph G, with each arc k having a weight $(\beta_k + \sum_{\ell \in \mathcal{L}_k} x_\ell)$. In summary, we solved formulation (5) by generating two types of constraints in a delayed fashion: Constraints (3d) and Constraints (5b). Because the separation problem for Constraints (5b) is a shortest path problem, which is much faster than the maximum weight independent set subproblem, we only generate Constraints (3d) if no more Constraints (5b) can be added.

Table 2 shows the optimal throughput for three values of the maximum number of hops. The first value, 14, is the minimum number of hops on a feasible path from one corner of the grid-7x7 dataset to the opposite. (The table also shows the runtime on a Dell laptop with a 2.70 GHz Intel i7 processor and 8 GHz of RAM.)

As Table 2 shows, the effect of the maximum number of hops was mostly insignificant. The only meaningful difference was for the grid-7x7 dataset with 3 jamming devices, indicating that having more possible routes can be beneficial when the network is subject to a more intense jamming attack. In addition, no consistent difference was noted in run time for different values of the maximum number of hops.

Table 2: Maximum number of hops vs. optimal throughput

Dataset	**Jamming Budget**	**Maximum number of hops**	**Runtime (s)**	**Optimal throughput**
cmu	1	14	168	0.500
		18	221	0.500
		24	174	0.500
	2	14	133	0.250
		18	117	0.250
		28	123	0.250
	3	14	33	0.000
		18	55	0.000
		28	38	0.000
grid-7x7	1	14	526	1.062
		18	638	1.065
		28	619	1.065
	2	14	1222	0.856
		18	1222	0.854
		28	1600	0.857
	3	14	1839	0.725
		18	2111	0.744
		28	1851	0.744

5.3 Runtime of Solution Procedure

Table 3 lists the runtimes for various datasets and problem instances. All experiments were run on compute nodes contained in a High Performance Computing Cluster using a 64-bit Linux operating system. A node has 2 Xeon X5670 Intel processors, which each have 8 cores and a clock speed of 2.93GHz and share 24GB of memory. The cutting plane algorithm was implemented with the Python interface for the Gurobi Optimizer [11], using the *LazyConstraints* feature. The maximum weight independent set separation problems were solved using Gurobi's branch-and-bound solver.

Table 3 shows the runtime for several solution procedures for several datasets. Each show shows the dataset, the number of channels (C), the number of jamming devices that can be located (R), and the run time for several procedures: branch-and-cut described in Section 4.1 (B&C), Benders decomposition described in Section 4.2.1 (Ben), and Benders decomposition implemented Gurobi's branch-and-cut algorithm using *LazyContraints* (BB&C). If the algorithm did not reach optimality after two hours, we report the percentage $Gap = (UB - LB)/LB$; we list "-" for cases in which no feasible solution was found after two hours.

Table 3: Runtime of various problem instances

Dataset	C	R	**Run time (s)**		
			B&C	**Ben**	**BB&C**
cmu	1	1	7	25	23
	2	1	30	79	74
	1	3	5	21	23
	2	3	32	120	208
grid-7x7	1	1	194	255	249
	2	1	349	577	932
	1	3	265	10%	2690
	2	3	4610	24%	81%
grid-8x8	1	1	428	1142	1156
	2	1	4333	10%	21%
	1	3	348	6795	161%
	2	3	57%	57%	282%
grid-9x9	1	1	800	2353	2962
	2	1	4708	22%	42%
	1	3	6065	73%	-
	2	3	-	88%	476%

As Table 3 shows, the branch-and-cut algorithm almost dominates the other algorithms investigated, with the exception being the grid-9x9 instance with $C = 2$ and $R = 3$. One reason for the success of branch-and-cut is that it adds a cut for every new maximum weight independent set found. Conversely, in the Benders decomposition algorithms tested, often many maximum weight

independent set problems must be solved to solve the Benders subproblem; therefore, the computational cost per cut is much higher. Thus, the branch-and-cut algorithm is analogous to a multi-cut version of Benders decomposition, which performs better than single-cut Benders decomposition on some problems. Another reason for the success of the branch-and-cut algorithm is that the Gurobi software may have been able to add more advanced cuts during the branch-and-bound procedure. On the whole, the classic Benders decomposition procedure (as described in Section 2) performs better than Benders implemented within branch-and-cut. We also added several modifications to the Benders decomposition procedures, such as Pareto optimality cuts [25] and knapsack inequalities (see Santoso et al. [39] for an example), but these additions actually increased the run time of the Benders decomposition procedures.

As Table 3 also shows, the run time generally increases with the number of nodes, as seen by comparing the runtimes of grid-7x7 instances with the runtime of their grid-8x8 and grid-9x9 counterparts. However, instances of the CMU dataset have significantly smaller runtimes than their grid-7x7 counterparts. This seems to be attributable to the fact that the transmitters are arranged differently in these two datasets. The runtime uniformly increases with an increase in the number of channels, which can be attributed to the fact that increasing the number of channels from 1 to 2 results in doubling the number of arcs in the network. The results show no consistent effect of the jamming budget on the runtime. (Note: the runtime of 7s for the CMU dataset with 1 channel and a jamming budget of 3 is small because in this case the minimum throughput is 0.0, making the instance easy to solve.)

In summary, the network size appears to be the most significant driver of runtime, followed by the number of channels. The capability of the algorithm presented in this paper is apparently reached for a 9x9 network (81 nodes).

5.4 Insights Into Designing a Network That is Robust Against Jamming Attacks

Strategy #1: Increase the Density of Nodes, Strategy #2: Increase the Communication Range, and Strategy #3: Decrease the Interference Range

When designing a wireless network, a decision-maker is able to decide on several characteristics. First, the designer can decide on the density of transmitters in the network. Second, the designer decides on the type of transmitter to use, which determines the communication range. In addition, we also investigate the strategy of decreasing the interference range of nodes, although this is not a real possibility in most cases.

Table 4 reports the optimal throughput for different sizes of datasets as well as different values of the communication range and interference range. Two values are used for the communication range: 1) the value 1/6, which is the range needed for every node to be able to communicate with its (up to four) nearest neighbors (North, South, East, West) in the grid-7x7 dataset and 2) $\sqrt{2}/6$, which is the range needed for any node in the grid-7x7 dataset to be able to communicate

to its nearest neighbors as well as any node diagonal to it (NE, SE, NW, SW). Three values of the interference range were used: 1) 0, which represents no interference, 2) the interference range is equal to communication range, and 3) the interference range is 1.75 times the communication range.

Table 4: Throughput vs. number of nodes, communication range, and interference range

Dataset	**Comm. range**	**Inf. range**	**Throughput**
cmu	1/6	0	0.5
cmu	1/6	1/6	0.3
cmu	1/6	0.29	0.3
cmu	$\sqrt{2}/6$	0	1.0
cmu	$\sqrt{2}/6$	1/6	0.7
cmu	$\sqrt{2}/6$	0.41	0.5
grid-7x7	1/6	0	2.0
grid-7x7	1/6	1/6	1.1
grid-7x7	1/6	0.29	0.9
grid-7x7	$\sqrt{2}/6$	0	2.0
grid-7x7	$\sqrt{2}/6$	1/6	1.0
grid-7x7	$\sqrt{2}/6$	0.41	0.6
grid-8x8	1/6	0	2.0
grid-8x8	1/6	1/6	1.3
grid-8x8	1/6	0.29	0.8
grid-8x8	$\sqrt{2}/6$	0	2.0
grid-8x8	$\sqrt{2}/6$	1/6	1.0
grid-8x8	$\sqrt{2}/6$	0.41	0.6
grid-9x9	1/6	0	2.0
grid-9x9	1/6	1/6	1.3
grid-9x9	1/6	0.29	0.6
grid-9x9	$\sqrt{2}/6$	0	2.0
grid-9x9	$\sqrt{2}/6$	1/6	1.0
grid-9x9	$\sqrt{2}/6$	0.41	0.5

The results displayed in Table 4 indicate, not surprisingly, that as the interference range increases, the throughput decreases (between 33% and 114%). Thus, a decision-maker could improve the throughput by selecting jamming devices that are more robust against interference.

One might expect that increasing the communication range would improve throughput. However, if the interference range is a multiple of the communication range (which is likely in practice), then increasing the the communication range actually *decreases* the optimal throughput for the grid networks. However, increasing the communication range does improve the throughput for the CMU dataset, perhaps because the variability in the inter-node distances make connectivity paramount.

The results also show that for the grid networks, increasing the density of nodes (e.g., moving from grid-7x7 to grid-8x8) may increase or decrease the optimal throughput. For no interference,

increasing the node density does not help. When the communication and interference ranges are both equal to 1/6, then increasing the size of the dataset does increase the optimal throughput, although this increase is less when moving from grid-8x8 to grid-9x9. On the other hand, when the interference range is 1.75 times the communication range (the default value), then the optimal throughput *decreases* with the density of nodes. This result varies with the observed behavior of wired network interdiction models, in which adding nodes typically improves the network's robustness to interdiction attacks because more redundant routes are added. This difference is likely due to the fact that, for a constant interference range and jamming range, adding additional nodes actually *accentuates* the effect of inference and jamming.

Finally, although the CMU and grid-7x7 datasets are of similar size (54 nodes vs. 49 nodes) the grid-7x7 dataset appears to perform much better during a jamming attack.

Strategy #4: Increase the Number of Channels

Another option in designing a wireless network is to include transmitters with multiple channels on which to send and receive signals. This naturally increases a network's robustness against jamming attacks because channels do not interfere with each other, meaning that adding channels results in many additional non-conflicting routes.

Table 5 shows how the throughput increases as more channels are added. The increase is significant, with the change ranging from 56% to 443%, although exhibiting diminishing returns. Indeed, utilizing channel hopping is a popular technique for resisting jamming attacks.

Table 5: Throughput vs. number of channels

Dataset	Number of Channels	Throughput	% Change
cmu	1	0.3	
	2	1.4	443%
	3	2.1	58%
grid-7x7	1	0.9	
	2	2.4	180%
	3	3.7	56%

Conclusions

In conclusion, increasing the number of channels appears to be the best strategy for designing a network that will be robust against jamming attacks. The improvement in the optimal throughput during a jamming attack is much more significant than the improvement for the strategy of reducing the interference range. Other strategies, such as increasing the density of transmitters and increasing the communication range, produced mixed results.

5.5 Insights Into Jamming a Network

Strategy #1: Increase the Number of Potential Jamming Locations

In the optimization model used in this paper, the jamming device placement decisions are represented by a binary variable. Thus, the area of possible jamming device locations must be discretized, turning the area into a finite set of points. With this construction, one must decide how many discrete points to include in the model. In this section, we analyze the effect of the number of potential jamming locations on the optimal throughput during a jamming attack.

As Table 6 shows, the number of jamming locations is usually insignificant, except for the CMU dataset when the number of locations is increased from 9 to 16 and then to 25. The fact that the throughput is relatively insensitive to the number of jamming locations bodes well for the jammer, who can decrease the number of jamming locations (decreasing the number of binary variables in the optimization model) without sacrificing solution quality.

Table 6: Throughput vs. number of potential jamming locations

Dataset	Number of jamming locations	Throughput
cmu	9	0.375
	16	0.286
	25	0.250
	36	0.250
	49	0.250
	64	0.250
grid_7x7	9	0.857
	16	0.857
	25	0.857
	36	0.857
	49	0.857
	65	0.857

Strategy #2: Increase the Number of Jamming Devices and Strategy #3: Increase the Jamming Interference Range

A jammer may also try to increase the the number of jamming devices, or use jamming devices with a larger range, in order to increase the efficacy of a jamming attack.

Table 7 reports how the optimal throughput changed in our experiments as a function of the number of jamming devices and jamming range. We tested two datasets of similar size: CMU (54 nodes) and grid-7x7 (49 nodes). Five values of the jamming range were used, each based on the dimensions of the grid-7x7 network. First, 0 was used, in which case the jammer can only simultaneously jam 1 device. Second, 1/12 was used, which is the jamming range needed to jam two transmitters. Third, the value of $\sqrt{2}/12 = 0.118$ was used, which is the range needed for the

device to jam 4 transmitters if placed at the center of a cell in the grid. Fourth, we used $1/6 = 0.166$, a range that allows a device to jam 3 transmitters if placed on a corner, 4 if placed on an edge, and 5 otherwise. Finally, we used $\sqrt{2}/6 = 0.235$, a range that allows a device to jam up to 9 transmitters.

Table 7: Throughput vs. number of jamming devices and jamming range

Number of jamming devices	**Jamming range**	**Throughput**	
		cmu	**grid-7x7**
1	0	0.79	1.07
1	1/12	0.50	1.07
1	$\sqrt{2}/12$	0.50	1.07
1	1/6	0.50	0.94
1	$\sqrt{2}/6$	0.25	0.91
2	0	0.79	0.86
2	1/12	0.25	0.86
2	$\sqrt{2}/12$	0.25	0.86
2	1/6	0.25	0.50
2	$\sqrt{2}/6$	0.00	0.33
3	0	0.79	0.75
3	1/12	0.00	0.75
3	$\sqrt{2}/12$	0.00	0.75
3	1/6	0.00	0.00
3	$\sqrt{2}/6$	0.00	0.00
4	0	0.79	0.50
4	1/12	0.00	0.50
4	$\sqrt{2}/12$	0.00	0.50
4	1/6	0.00	0.00
4	$\sqrt{2}/6$	0.00	0.00
5	0	0.79	0.00
5	1/12	0.00	0.00
5	$\sqrt{2}/12$	0.00	0.00
5	1/6	0.00	0.00
5	$\sqrt{2}/6$	0.00	0.00

As Table 7 shows, increasing the jamming range monotonically decreases the optimal throughput. Increasing the jamming range by one level caused the optimal throughput to reduce by between 25% and 100%. No difference in this effect is apparent between the two datasets.

In addition, increasing the jamming budget also decreases the optimal throughput. The decrease for the CMU dataset is always either 25% or 0%. For the grid-7x7 dataset, the decrease was between 0% and 57%. This significant difference could be due to the fact that the 2-D grid arrangement of the transmitters in the grid-7x7 network enables a jamming attack to disconnect the network more

easily than if the transmitters were arranged without a pattern. Thus, while the grid-7x7 performs better during an attack consisting of 2 jamming devices (see Strategies 1, 2, and 3) in the previous section, overall the CMU network is more robust against a jamming attack.

Conclusions

The most beneficial strategy for a jammer interested in minimizing the network throughput during a jamming attack would appear to be to increase the range of the jamming devices. Increasing the number of devices also is beneficial.

6 Conclusions

6.1 Discussion

Wireless network security is currently an important topic, mostly because 1) wireless networks are ubiquitous and 2) the wireless network medium is inherently vulnerable to attacks. A sub-area of wireless network security is *wireless network jamming*, which has been studied by many in the electrical engineering community but by few in the operations research community. In this paper, we addressed one problem in wireless network jamming, namely the problem of where to place jamming devices in order to minimize the throughput of a wireless network. Our paper is the first to consider two important aspects of this problem: 1) wireless networks are subject to interference and 2) network throughput is used as the optimization objective.

We solved this problem by formulating it as a mixed-integer bi-level program with an exponential number of constraints in the inner problem. After using the standard approach of taking the inner dual, we used a cutting plane approach to solve the bi-level problem to optimality. This cutting plane approach, which was found to be superior to a Benders decomposition approach, was able to solve networks of up to 81 nodes to optimality. Thus, we believe that this approach could serve as a good foundation to build upon as others solve new problems in wireless network jamming subject to interference.

We used our model and algorithm in a series of experiments with the goal of answering three questions:

1. *Does constraining the number of hops allowed on a path decrease the optimal throughput?* We did not find a significant relationship between the number of hops allowed and the optimal throughput.

2. *What strategies for designing a network to be jamming-resistant are most beneficial?* We found that increasing the number of channels appears to be the best strategy for designing a network that is robust against jamming attacks. The benefit from this strategy is much more significant than the benefit from reducing the interference range. Other strategies such as increasing the density of transmitters and increasing the communication range produced mixed results.

3. *What strategies for maximizing the impact of a jamming attack are most beneficial?* We found the most beneficial strategy is to increase the range of the jamming devices. Increasing the number of devices is also beneficial.

6.2 Future Work

Wireless network jamming is a fertile ground for interesting optimization research, and yet this area has not received much attention by the optimization community. Thus, many potentially fruitful avenues exist for future research. The most obvious is to study how to design an optimal wireless network that can optimize network performance during a jamming attack. Design decisions include the layout of the transmitters, the amount of power allocated to each transmitter, the number of channels, and the number of radios available at each transmitter. Next, although we focused on the *protocol model* of interference, the *physical model* should also be studied from the standpoint of interdiction. The cutting plane procedure used in this paper would still be valid, except that rather than solving a maximum weight independent set problem, the separation problem would be a maximum weight schedulable set problem [16], which is likely more difficult to solve.

References

[1] Ahuja, R. K., Magnanti, T. L., Orlin, J. B., Feb. 1993. Network Flows: Theory, Algorithms, and Applications, 1st Edition. Prentice Hall, Upper Saddle River, NJ.

[2] Arulselvan, A., Commander, C. W., Elefteriadou, L., Pardalos, P. M., Jul. 2009. Detecting critical nodes in sparse graphs. Computers & Operations Research 36 (7), 2193–2200.

[3] Awerbuch, B., Richa, A., Scheideler, C., 2008. A jamming-resistant MAC protocol for single-hop wireless networks. In: Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing. PODC '08. ACM, New York, NY, USA, pp. 45–54.

[4] Bayraktaroglu, E., King, C., Liu, X., Noubir, G., Rajaraman, R., Thapa, B., Oct. 2013. Performance of IEEE 802.11 under jamming. Mob. Netw. Appl. 18 (5), 678–696.

[5] Bodik, P., Guestrin, C., Hong, W., Madden, S., Paskin, M., Thibaux, R., 2004. Intel lab data.

[6] Brown, G. G., Carlyle, W. M., Harney, R. C., Skroch, E. M., Wood, R. K., 2009. Interdicting a Nuclear-Weapons Project. Operations Research 57 (4), 866.

[7] Commander, C., Pardalos, P., Ryabchenko, V., Shylo, O., Uryasev, S., Zrazhevsky, G., Jan. 2008. Jamming communication networks under complete uncertainty. Optimization Letters 2 (1), 53–70.

[8] Commander, C. W., Pardalos, P. M., Ryabchenko, V., Uryasev, S., Zrazhevsky, G., 2007. The wireless network jamming problem. Journal of Combinatorial Optimization 14 (4), 481–98.

[9] Cormican, K. J., Morton, D. P., Wood, R. K., Feb. 1998. Stochastic network interdiction. Oper. Res. 46 (2), 184–197.

[10] Du, W., Deng, J., Han, Y. S., Chen, S., Varshney, P. K., Mar. 2004. A key management scheme for wireless sensor networks using deployment knowledge. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. Vol. 1. IEEE, p. 597.

[11] Gurobi Optimization, I., 2014. Gurobi optimizer reference manual.

[12] Held, H., Hemmecke, R., Woodruff, D. L., 2005. A decomposition algorithm applied to planning the interdiction of stochastic networks. Naval Research Logistics 52 (4), 321–328.

[13] Held, H., Woodruff, D. L., 2005. Heuristics for multi-stage interdiction of stochastic networks. Journal of Heuristics 11 (5-6), 483–500.

[14] Israeli, E., Wood, R. K., Sep. 2002. Shortest-Path network interdiction. Networks 40 (2), 97–111.

[15] Iyer, A., Rosenberg, C., Karnik, A., May 2009. What is the right model for wireless channel interference? IEEE Transactions on Wireless Communications 8 (5), 2662–2671.

[16] Jain, K., Padhye, J., Padmanabhan, V. N., Qiu, L., Jul. 2005. Impact of interference on Multi-Hop wireless network performance. Wireless Networks 11 (4), 471–487.

[17] Jiang, S., Xue, Y., Aug. 2009. Optimal wireless network restoration under jamming attack. In: Proceedings of 18th Internatonal Conference on Computer Communications and Networks, ICCCN 2009. IEEE, pp. 1–6.

[18] Law, Y. W., van Hoesel, L., Doumen, J., Hartel, P., Havinga, P., 2005. Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. In: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks. SASN '05. ACM, New York, NY, pp. 76–88.

[19] Lei, T. L., Apr. 2013. Identifying critical facilities in Hub-and-Spoke networks: A hub interdiction median problem. Geogr Anal 45 (2), 105–122.

[20] Li, M., Koutsopoulos, I., Poovendran, R., May 2007. Optimal jamming attacks and network defense policies in wireless sensor networks. In: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE. IEEE, pp. 1307–1315.

[21] Lim, C., Smith, J. C., Jan. 2007. Algorithms for discrete and continuous multicommodity flow network interdiction problems. IIE Transactions 39 (1), 15–26.

[22] Liu, Y., Ning, P., Dai, H., Liu, A., Mar. 2010. Randomized differential DSSS: Jamming-Resistant wireless broadcast communication. In: INFOCOM 2010. IEEE, pp. 1–9.

[23] Lunday, B. J., Sherali, H. D., Jan. 2010. A dynamic network interdiction problem. Informatica 21 (4), 553–574.

[24] Ma, K., Zhang, Y., Trappe, W., Nov. 2005. Mobile network management and robust spatial retreats via network dynamics. In: IEEE International Conference on Mobile Adhoc and Sensor Systems Conference. IEEE, pp. 235–242.

[25] Magnanti, T. L., Wong, R. T., 1981. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. Operations Research 29 (3), 464–484.

[26] Malaviya, A., Rainwater, C., Sharkey, T., Jul. 2012. Multi-period network interdiction problems with applications to city-level drug enforcement. IIE Transactions 44 (5), 368–380.

[27] Morton, D. P., Pan, F., Saeger, K. J., 2007. Models for nuclear smuggling interdiction. IIE Transactions 38 (1), 3–14.

[28] Navda, V., Bohra, A., Ganguly, S., Rubenstein, D., May 2007. Using channel hopping to increase 802.11 resilience to jamming attacks. In: 26th IEEE International Conference on Computer Communications, INFOCOM. IEEE, pp. 2526–2530.

[29] Noubir, G., 2004. On connectivity in ad hoc networks under jamming using directional antennas and mobility. In: Langendoerfer, P., Liu, M., Matta, I., Tsaoussidis, V. (Eds.), Wired/Wireless Internet Communications. Vol. 2957 of Lecture Notes in Computer Science. Springer, pp. 186–200.

[30] Noubir, G., Lin, G., Jul. 2003. Low-power DoS attacks in data wireless LANs and countermeasures. SIGMOBILE Mob. Comput. Commun. Rev. 7 (3), 29–30.

[31] Pan, F., Charlton, W., Morton, D. P., 2003. Interdicting smuggled nuclear material. In: Woodruff, D. L. (Ed.), Network Interdiction and Stochastic Integer Programming. Kluwer Academic Publishers, Boston, pp. 1–20.

[32] Pan, F., Morton, D. P., Oct. 2008. Minimizing a stochastic maximum-reliability path. Networks 52 (3), 111–119.

[33] Panyim, K., Hayajneh, T., Krishnamurthy, P., Tipper, D., Oct. 2009. On limited-range strategic/random jamming attacks in wireless ad hoc networks. In: Proceedings of the 34th IEEE Conference on Local Computer Networks, LCN. IEEE, pp. 922–929.

[34] Peixoto, T. P., 2014. The graph-tool python library. figshare.

[35] Pelechrinis, K., Iliofotou, M., Krishnamurthy, S. V., 2011. Denial of service attacks in wireless networks: The case of jammers. IEEE Communications Surveys and Tutorials 13 (2), 245–257.

[36] Pelechrinis, K., Koutsopoulos, I., Broustis, I., Krishnamurthy, S. V., Nov. 2009. Lightweight jammer localization in wireless networks: System design and implementation. In: Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM. IEEE, pp. 1–6.

[37] Richa, A., Scheideler, C., Schmid, S., Zhang, J., Jun. 2011. Competitive and fair medium access despite reactive jamming. In: Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 507–516.

[38] Salmeron, J., Wood, K., Baldick, R., 2009. Worst-Case Interdiction Analysis of Large-Scale Electric Power Grids. IEEE Transactions on Power Systems 24 (1), 96–104.

[39] Santoso, T., Ahmed, S., Goetschalckx, M., Shapiro, A., 2005. A stochastic programming approach for supply chain network design under uncertainty. European J. Oper. Res. 167 (1), 96–115.

[40] Shen, S., Smith, J. C., Sep. 2012. Polynomial-time algorithms for solving a class of critical node problems on trees and series-parallel graphs. Networks 60 (2), 103–119.

[41] Tague, P., Slater, D., Poovendran, R., Noubir, G., Apr. 2008. Linear programming models for jamming attacks on network traffic flows. In: Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops. IEEE, pp. 207–216.

[42] Wood, Jan. 1993. Deterministic network interdiction. Mathematical and Computer Modelling 17 (2), 1–18.