

Avoiding High Impacts of Geospatial Events in Mission Critical and Emergency Networks using Linear and Swarm Optimization

M. Todd Gardner

Federal Aviation Administration
Kansas City, Missouri
todd.gardner@faa.gov

Cory Beard, Deep Medhi

Computer Science Electrical Engineering
University of Missouri-Kansas City
Kansas City, Missouri 64110-2499
beardc@umkc.edu, dmedhi@umkc.edu

Abstract— Geospatial events continue to plague both wireless and wireline communication networks. The immediate effect of a large scale geospatial event is generally complete or partial loss of situational awareness caused by a lack of communications and media availability in the affected area, effectively isolating many people affected. Significant challenges have been reported by emergency responders, victims, and other involved persons immediately following the recent U.S. tornados and the earthquakes worldwide. Riots and protests can also act as geospatial events that stress communications resources in a geographic area.

This research develops novel optimization models to identify and mitigate geospatial vulnerabilities in network designs before they occur. We use an integer linear program (ILP) to add nodes that reduces geographic vulnerability thus preventing users from being isolated by geospatial events. To expand the scope to include more solutions and a wider range of objective functions, a swarm optimization approach was also developed. Wireless propagation models that include obstructions like buildings and other terrain features are tested with these models as well.

Keywords- situation management, geo-spatial planning, emergency management, geographic vulnerability, swarm optimization, linear programming, network planning, topology design

I. INTRODUCTION

During a disaster, communications systems can be devastated. When this occurs, a communications “black hole” is created at the center of the disaster. This causes a significant lack of *situational awareness* immediately that extends throughout the geographic impact zone. The network infrastructure that remains becomes stressed from a flood of users trying to communicate. This is the basic description of a *geospatial* event in a network. Events of this nature are not limited to small or large geographic areas. In addition, the network becomes “mission critical” for survivors and responders. [1] and [2] discuss the criticality of the communications following a disaster.

Typically network designers separate topology from geography. In this work we develop optimization models to add nodes to a network topology that eliminate the geographic vulnerabilities, protecting the network from geospatial events.

We develop models based on Integer Linear Programming (ILP) and Particle Swarm Optimization (PSO) to optimize networks for both point-to-point and all-terminal applications. To demonstrate the flexibility and applicability of both approaches, we use simple wireless propagation link models as well as more complex link models that include obstructions like buildings and other terrain features.

Both the ILP based approach and the PSO based approach were successful in mitigating geographic vulnerabilities found in our test networks. We use a geographic vulnerability function $V(N,E,r)$ that was developed in [3] and is defined in Appendix 1 to evaluate our solutions and participate in the objective function of the PSO approach. Since the $V(N,E,r)$ is non-linear, the ILP approach is only an approximation. The PSO approach can use $V(N,E,r)$ as the objective providing a more direct solution. Another important difference between the two approaches is that the ILP approach can augment the network with multiple nodes concurrently, whereas the PSO approach adds one node at a time.

A. Motivation

During the Japan (Tohoku) Earthquake, millions of telephone lines were damaged. Wireless communications were poor in the affected areas in Japan. In many cases, texting, social media, and Skype were the only form of communications for survivors [4][5]. During the May 21, 2011 tornado in the U.S, phone communications to the city of Joplin were largely cutoff after the devastating storm cut a 6.5 mile long and one-half mile wide path through the town. Interestingly enough, as storm chasers broadcasted live from the tornado’s path, people that lived hundreds of miles away were able to understand the scope of the disaster better than people living in the middle of the tornado’s path that had no phone or television service [6]. The immediate lack of *situational awareness* caused survivors to use whatever means available, which frequently included texting, Twitter, and Facebook, to communicate their situation to the outside world. Since voice communications is typically prioritized for emergency response personnel during disasters [7], any connection (even text) to the network becomes vital for survivors. Another concern are secondary events that are related to the original event but not necessarily predictable.

Examples are secondary flooding after the initial event, movement of fires (forest, structure), earthquake aftershocks, structure collapse/explosions (from natural gas leaks, structure instability).

What these disasters bring to light is the need for communications networks that are resilient to geospatial events. It is our contention that communications networks should not be vulnerable to geospatial failures.

B. Contributions

In [3], Geographic Vulnerability is defined as follows

DEFINITION 1 (GEOGRAPHIC VULNERABILITY). *A Geographic Vulnerability is the geographic area of a network that if attacked can cause significant impact to the function of the entire network.*

DEFINITION 2 (GEOSPATIAL EVENT). *A Geospatial Event is an event that can cause a geographic vulnerability.*

DEFINITION 3 (THREAT RADIUS). *The Threat Radius is the physical radius of a geospatial event that is used to define the geographic vulnerability.*

Figure 1 shows an example of geographic vulnerabilities in a 20 node network. Note that when multiple nodes need to be disabled concurrently, the vulnerable area is the intersection of the threat radius around those nodes [3]. With this in mind, we try to accomplish the following with this work:

- Develop an Integer Linear Program (ILP) that augments a network by selecting locations to add nodes to a network which *approximates* the minimization of the geographic vulnerability $V(N,E,r)$ of that network. This applies both to point-to-point demands as well as point-to-any (all-terminal) demands. For point-to-point demands we create paths that are physically diverse from a line between the source and destination (s,t) . New nodes are added (with a new node cost) that lower the diversity cost of the path. The all-terminal model is similar, using spanning trees as opposed to paths.
- Develop a particle swarm optimization (PSO) approach to select a location to add a single node that minimizes the geographic vulnerability $V(N,E,r)$ for the point-to-point case and the all-terminal case. Because PSO is used, non-linear objective functions can be employed. The objective function used is $V(N,E,r)$ as described in Appendix 1.
- Test both approaches (ILP and PSO) with networks of various sizes (15 nodes to 50 nodes) and multiple link models, allowing our approaches to be used in wireless networks as well as wired networks. In addition, we include obstructions in the link models.

Section II considers existing survivable network design techniques. Section III describes our methods to solve the geographic vulnerability problem. Finally, we present our results and conclusions.

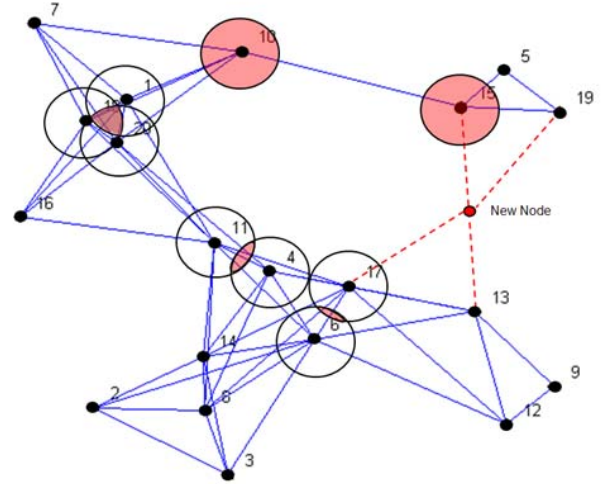


Figure 1. Wireless Network showing Geographic Vulnerabilities and a New Node Location that Eliminates the Vulnerabilities.

II. RELATED WORK

There have been significant recent advances in the assessment of geographically correlated failures and network design related to geographically correlated vulnerabilities.

Gardner and Beard [3] developed geographic vulnerability assessment methods as discussed in Appendix 1. Naumayer and Modiano [10] completed work that models disasters as line segments or disks and calculates worst case network link cuts based on the line (or disk) intersecting with the links. Our work considers geographically correlated node failures as opposed to geographically correlated link failures.

Sen, Murthy, and Banerjee in [13] construct region disjoint paths between a source and destination. Their work has some similarities to our work. The regions in their work are predetermined. Li, Wang, and Jiang [12] expand on the work in [13] in two areas. First, they consider multiple regions as opposed to one. Second, they augment the network by adding link capacity to ensure all traffic can withstand multiple region failures. Instead of predetermining regions, they assume regions are centered at each node. Our work designs the network assuming the region can be centered anywhere in the network. Also, we augment our network with new node locations as opposed to finding region disjoint paths within existing nodes.

In [11], Cetinkaya, Broyles, Dandekar, Srinivasan, and Sterbenz have taken a different approach and developed a simulation platform that is capable of locating geographic vulnerabilities in networks. Their framework is capable of modeling both malicious and non-malicious attacks, disasters, and wireless challenges.

III. GEOSPATIAL NETWORK DESIGN

Our goal with this work is to find approaches that augment networks to reduce or eliminate geographic vulnerabilities or to *Minimize* $V(N,E,r)$. In Appendix 1, $V(N,E,r)$ is defined as the Geographic Vulnerability of a network with nodes N , links E , and threat radius r .

The solution is more complex because $V(N,E,r)$ is a non-linear function. We chose two approaches. The first approach an Integer Linear Program (ILP) to *approximate* the minimization of $V(N,E,r)$. We use the ILP approach to provision two paths from the source to the destination (point-to-point case). If the two paths are physically separated by at least the threat radius, the effect is to reduce $V(N,E,r)$ as shown in Figure 2. Both solutions are feasible. But, path 2 reduces $V(N,E,r)$ more than path 1 due to nodes 10 and 15.

Next, we use a PSO approach which allows us to minimize the $V(N,E,r)$ as our objective function. PSO particles move across the network space using the PSO algorithm, testing the objective at each particle location as they move. The particle location that minimizes the objective function the most is added as a new node.

A. Integer Linear Program (ILP) Based Approach

The ILP formulation is based on the premise that the program is executed two times with opposite weighting each time, creating two diverse paths (point-to-point) or two diverse spanning trees (all-terminal). Since calculating geographic vulnerability $V(N,E,r)$ is an inherently non-linear problem, we use weights that make it more costly for the diverse paths or diverse spanning trees to be geographically close together. The “geographic space” in which the network resides is weighted to push Path 1 or Spanning Tree 1 to a specific geographic area. The weights are then changed to push Path 2 or Spanning Tree 2 to the opposite geographic area. Possible new nodes are selected from a grid pattern with a “new node” cost to improve a poorly weighted path.

By ensuring that two paths exist through the network on opposite sides of the network, vulnerabilities located anywhere in the network can be mitigated. A single path may not mitigate the vulnerability, if it is located in the same area as the path is weighted toward as shown in Figure 2. The same rational applies for using spanning trees located in different physical areas of the network. Path 1 or Path 2 (Spanning Tree 1 or 2) may then be chosen to augment the network.

Spanning trees are used in the all-terminal case ensuring that all nodes can communicate with all other nodes. We chose nodes with the lowest geographic diversity cost to serve as root

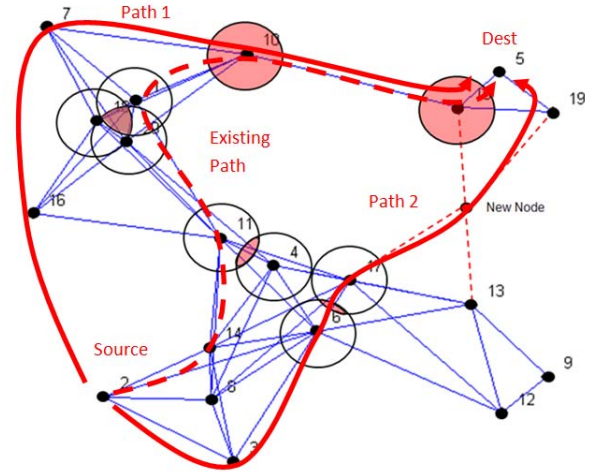


Figure 2. Point-to-Point Scenario Showing Two Potential Diverse Paths on Opposite Sides of the Network.

nodes. This prevents poor root node choice from adding significant diversity cost to the optimization. Since, we are concerned only with connectivity, any spanning tree will suffice.

Figure 3 shows the node weights for an example point-to-point case. The weighting scheme is based on the perpendicular distance from a line across the geographic space that includes the source and destination nodes (s,t). For Path 1, the weights are normalized at zero on the furthest edge perpendicular to the source-destination line and grow toward the opposite side of the geographic space. For Path 2, the weights are reversed with zero being normalized to the opposite corner and growing the other direction. The effect is that Paths 1 and 2 would form on opposite sides of the network space geographically.

For the all-terminal example shown in Figure 4, the weighting scheme is based on an interior/exterior concept. For Spanning Tree 1, the weights start at the center at zero and grow toward the exterior of the space based on the distance from the center of the space. The weighting scheme for

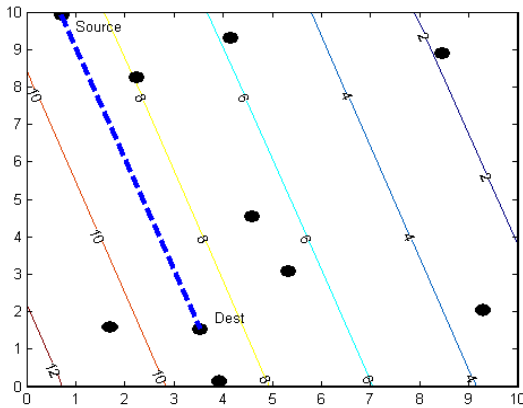


Figure 3. Point-to-Point Weighting Plan (Path 1) for an Example 10 Node Network.

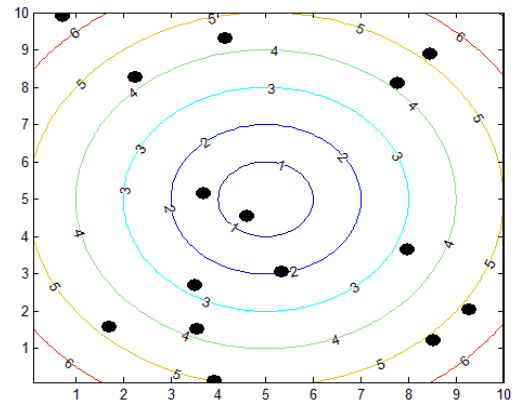


Figure 4. All-Terminal Weighting Plan (Spanning Tree 1) for an Example 15 Node Network ($\alpha = 1$).

Spanning Tree 2 will have the furthest point in the space starting at zero and growing toward the center. Spanning Tree 1 will tend to form in the center of the network space and Spanning Tree 2 will tend to form on the edge of the graph providing diversity between the two trees.

$$W = d^\alpha \quad \alpha > 0 \quad (2)$$

Equation (2) shows the calculation of the geographic diversity cost W . α is a constant used to scale the weights exponentially. d is the distance from the (s,t) line (source - destination) or from the center for the all-terminal case. The ILP approach uses a node-link formulation as described in [14] that includes all nodes and links (existing and potential). The formulation follows.

Notation

$e = 1, 2, \dots, E$ List of all possible links
 $n = 1, 2, \dots, m-1, m \dots N$ List of all existing $(1 \dots (m-1))$ and potential new $(m \dots N)$ nodes

Constants

W_n Diversity Cost at node n
 a_{en} Used for Node-Link formulation, 1 if node n is the originating node of link e ; else 0
 b_{en} Used for Node-Link formulation, 1 if node n is the terminating node of link e ; else 0
 s Source node of demand
 t Sink node of demand
 N Number of nodes (existing and possible)
 C Cost to add a node
 K Maximum number of possible added nodes (optional)

Variables

u_e Binary variable is '1' if link e is used, else 0
 v_n Integer variable is indicative of traffic flows passing through the node.

Objective

$$\min \sum_n W_n v_n + C \sum_p v_p \quad n = 1 \dots N; \quad (3)$$

$p = m \dots N$ (potential new nodes)

Constraints (point-to-point)

$$\sum_e a_{en} u_e - \sum_e b_{en} u_e = \begin{cases} 1 & \text{if } n = s \\ 0 & \text{if } n \neq s, t \\ -1 & \text{if } n = t \end{cases} \quad (4)$$

$$n = 1, 2, \dots, N \quad (5)$$

$$\sum_e a_{en} u_e = v_n \quad n = 1, 2, \dots, N$$

Equation (4) is the balance equation for a node-link formulation [14]. Equation (5) sets v_n to the units of flow passing through node n . In the point-to-point model, the total flow is '1', v_n is either '1' or '0'.

Constraints (all-terminal)

$$\sum_e a_{en} u_e - \sum_e b_{en} u_e = \begin{cases} N-1 & \text{if } n = \text{root} \\ -1 & \text{for all others} \end{cases} \quad (6)$$

$$n = 1, 2, \dots, N$$

$$\sum_e a_{en} u_e = v_n \quad n = 1, 2, \dots, N \quad (7)$$

Equation (6) is modified to create a spanning tree. A root node is chosen by finding the least weighted existing node. This node is the source for one unit of traffic to every other node. Equations (6) and (7) serve the same function as (4) and (5). One interesting note is that in the All-Terminal model, v_n represents the number of other nodes accessed through that node. A busy node is penalized more than a less busy node. For the point-to-point case, Equation (8) that can be added as a constraint to limit the number of new nodes added.

$$\sum_n v_n \leq K \quad n = m \dots N (\text{added nodes only}) \quad (8)$$

For the point-to-point and all-terminal models, this ILP has $[2N]$ constraints and $[N(N-1)+N]$ variables.

B. Swarm Optimization

This work uses the Full Particle Swarm Optimization (PSO) model as described in [15] to choose nodes to add to the network to improve the network resilience to geographic failures. The PSO formulation is based on the objective of minimizing the geographic vulnerability function $V(N, E, r)$. P particles are placed in the network physical space. Each particle represents a potential location to add as a node. A grid pattern is used in this work to spread the starting points for possible solutions. Other particle starting patterns (like random) could easily be used. The size of P is also configurable.

Equation (9) is the PSO position equation. $x_{ij}(t)$ is the current particle location with particle i representing a possible solution in dimension j . This is considered a two dimensional problem, with the dimensions in the x and y directions. $x_{ij}(t+1)$ is the next position. $v_{ij}(t+1)$ is the next velocity. t is the iteration number.

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (9)$$

Equation (10) is the PSO velocity equation. $y_{ij}(t)$ is the particle i best position (based on the objective) in dimension j from the start till the current time t . $\hat{y}_j(t)$ is the global (all particles) best position in dimension j . c_1 and c_2 are the cognitive and social acceleration constants. $r_1(t)$ and $r_2(t)$ are uniform random numbers in from $[0, 1]$.

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_1(t) [y_{ij}(t) - x_{ij}(t)] + c_2 r_2(t) [\hat{y}_j(t) - x_{ij}(t)] \quad (10)$$

Finally, (11) shows how $y_i(t+1)$ is calculated. Generally, for a given particle if $f(x_i(t+1)) = V(N, E, r)$ is a better solution than $f(y_i(t))$, then the new personal best $y_i(t+1)$ is set to $x_i(t+1)$. In addition, if the best $y_i(t+1)$ is better than $\hat{y}_j(t)$ then $\hat{y}_j(t) = y_i(t+1)$.

TABLE I. TEST NETWORK CONFIGURATION

Networks A 1-3 Configuration		
Nodes	15	
Wireless TX Level	15	dBm
Antenna Gain	0	dB
Receive Threshold	-98	dBm
Frequency (MHz)	2400	MHz
Link Generation Method	FSL	
Obstruction loss	0	dB
Networks B 1-3 Configuration		
Nodes	25	
Wireless TX Level	25	dBm
Antenna gain	10	dB
Receive Threshold	-120	dBm
Frequency	1000	MHz
Link Generation Method	ITU-R	
Obstruction loss	6	dB
Networks C 1-3 Configuration		
Nodes	50	
Wireless TX Level	18	dBm
Antenna gain	10	dB
Receive Threshold	-120	dBm
Frequency	1000	MHz
Link Generation Method	ITU-R	
Obstruction loss	6	dB

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad (11)$$

To compute $f(x_i(t+1)) = V(N,E,r)$ in a network of N nodes:

1. Convert location of particle $x_i(t+1)$ to new node
2. Add node to the set of nodes N
3. Generate set of links E based on set of nodes N .
4. Compute $V(N,E,r)$ as described in Appendix 1.

These stopping parameters halt PSO movement.

1. A geographic vulnerability threshold is reached.
2. Maximum iterations are reached.

C. Link Generation Techniques

To demonstrate the model usage in realistic scenarios, we tested different wireless link generation models that could

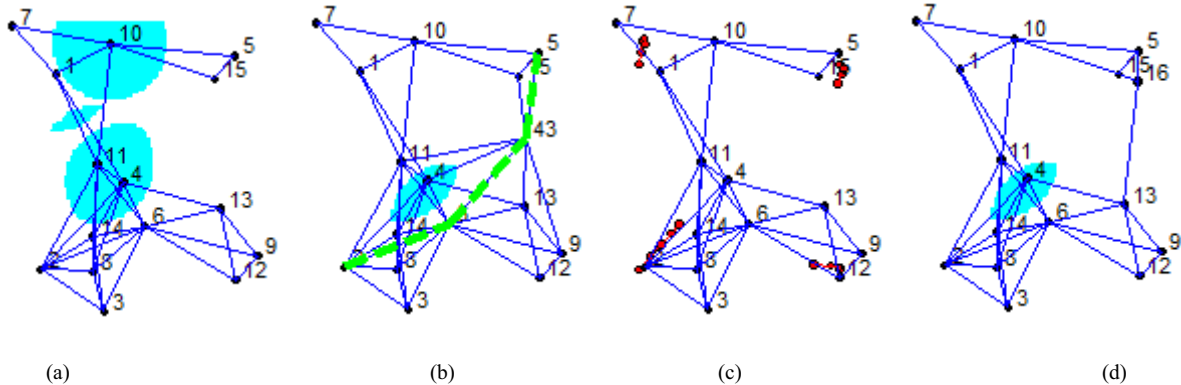


Figure 5. Network A1 Point-to-Point Example. (a) Network A1 Showing the $V(N,E,r)$ of 0.182 prior to Augmentation. (b) ILP Optimized Showing Path 1 (Added Node 43), $V(N,E,r)$ of 0.028. (c) PSO movement. (d) PSO Optimized Network (Added Node 16), $V(N,E,r)$ of 0.028.

TABLE II. $V(N,E,r)$ - POINT-TO-POINT

Network	Initial	ILP Optimized		Swarm	
A1	0.182	0.028	(1)	0.028	(1)
A2	0.116	0.018	(1)	0.029	(1)
A3	0.063	0.000	(1)	0.009	(1)
B1	0.183	0.000	(1)	0.000	(1)
B2	0.333	0.000	(1)	0.000	(1)
B3	0.216	0.000	(1)	0.043	(1)
C1	0.062	0.000	(4)	0.021	(1)
C2	0.204	0.000	(1)	0.031	(1)
C3	0.075	0.017	(2)	0.024	(1)
Number of nodes added are shown in parentheses					

TABLE III. $V(N,E,r)$ - ALL-TERMINAL

Network	Initial	ILP Optimized		Swarm	
A1	0.236	0.122, 0.000	(1,6)	0.070	(1)
A2	0.278	0.152, 0.000	(1,7)	0.141	(1)
A3	0.089	0.012, 0.000	(2,4)	0.019	(1)
B1	0.356	0.090, 0.017	(2,4)	0.082	(1)
B2	0.449	0.049, 0.004	(2,6)	0.049	(1)
B3	0.230	0.092, 0.000	(1,4)	0.047	(1)
C1	0.341	0.099, 0.0551	(3,7)	0.147	(1)
C2	0.329	0.133, 0.0146	(3,4)	0.091	(1)
C3	0.301	0.156, 0.0275	(3,11)	0.211	(1)
1. Number of nodes added are shown in parentheses					
2. ILP Optimized shown for 'low' and 'high' weight option					

include terrain obstructions. Two basic models were used to generate links.

1. Free Space Loss (FSL). Friis's Law is used to determine if sufficient power is available at the receiving node [16].
2. ITU-R Pedestrian Model with Obstructions. Equation (12) shows the ITU-R Pedestrian Model [16]. A 6 dB loss is added if the path crosses obstructions.

$$PL = 40 \log_{10} d + 30 \log_{10} f_c + 49 \quad (12)$$

IV. RESULTS

To evaluate the methods in this paper, we have selected 9 different randomly generated networks, three 15 node (A1 – A3), three 25 node (B1 – B3), and three 50 node (C1 – C3) networks. In networks A1-A3, we assumed no obstructions and the free space loss (FSL) link model. For networks B1-B3 and C1-C3, we chose to include obstructions and used the ITU-

R Pedestrian Model to calculate loss. All networks are assumed to occupy a 10 km x 10 km space. The configuration parameters are shown in Table I.

The ILP parameters needed to add nodes to a network are *strongly* dependant on the topology of that individual network. For the point-to-point case C was set to 1.0 and α was set to 0.5 for networks A(1-3) and B(1-3). For network C(1-3), α was varied from 0.25 and 1.0 till nodes were added to the solution. The same approach was used for the all-terminal case. In the high weight option, α ranges from 0.5 to 0.75. C remained 1.0. In the low weight option, C ranges from 2.0 to 5.0. The PSO approach used acceleration constants of $c1$ and $c2$ of 0.1 for all cases. Stopping parameters of 16 iterations and $V(N,E,r) = 0.03$ were used for all tests.

In Tables II and III, we see the geographic vulnerability $V(N,E,r)$ of the A, B, and C networks with no augmentation, augmentation with the ILP approach, and augmentation with the PSO approach. It is interesting to note that the ILP approach can add multiple nodes. This occurs frequently because one node (in the predetermined location) may not sufficiently reduce the diversity cost to bring that solution into feasibility. Since the diversity cost is an approximation of $V(N,E,r)$, nodes can be added that reduce the diversity cost but do not improve $V(N,E,r)$. The ILP approach can be weighted such that nodes are cheap and many new nodes are brought into feasibility or such that nodes are expensive. This will create varying values of $V(N,E,r)$.

The PSO approach has the ability to find the optimum solution, but since it is based partially on random movement, it is possible to find a good solution but not the “best” solution. The all-terminal scenario presents interesting challenges because normally there are multiple unrelated vulnerabilities in the network that need to be mitigated.

A. Network A1 – Point-to-Point Scenario

Figure 5 shows Network A1 (the point-to-point scenario). The source node is 2 and the destination node is 5. Figure 5(a) shows the geographic vulnerabilities in this network for the 2-5 pair. It is clear why the vulnerabilities are centered on nodes 10, 11&1, and 11&4. The vulnerability radius used is 2.

In Figure 5(b), the path formed by weighting the network to reduce the diversity cost to the right of the line from 2-5. The cost of a new node is $C = 1$ and $\alpha = 0.5$. Node 43 is added which eliminates the vulnerabilities caused by node 10, 11&4, 11&1, but leaves a smaller vulnerability caused by nodes 11&6. Figure 5(c) shows the 4 initial swarm particle locations. The particle movement is shown. The vulnerability (0.028) was found and the movement stopped after a threshold of (0.03) was reached as shown in Figure 5(d).

B. Network B1 – All-Terminal Case

Figure 6 shows a 25 node example (Network B1) for the all terminal case. In Figure 6(a), we see the connectivity generated by the ITU-R Pedestrian Model with a random set of obstructions as shown. Each obstruction exterior wall contributes 6dB to a path. The non-optimized network in

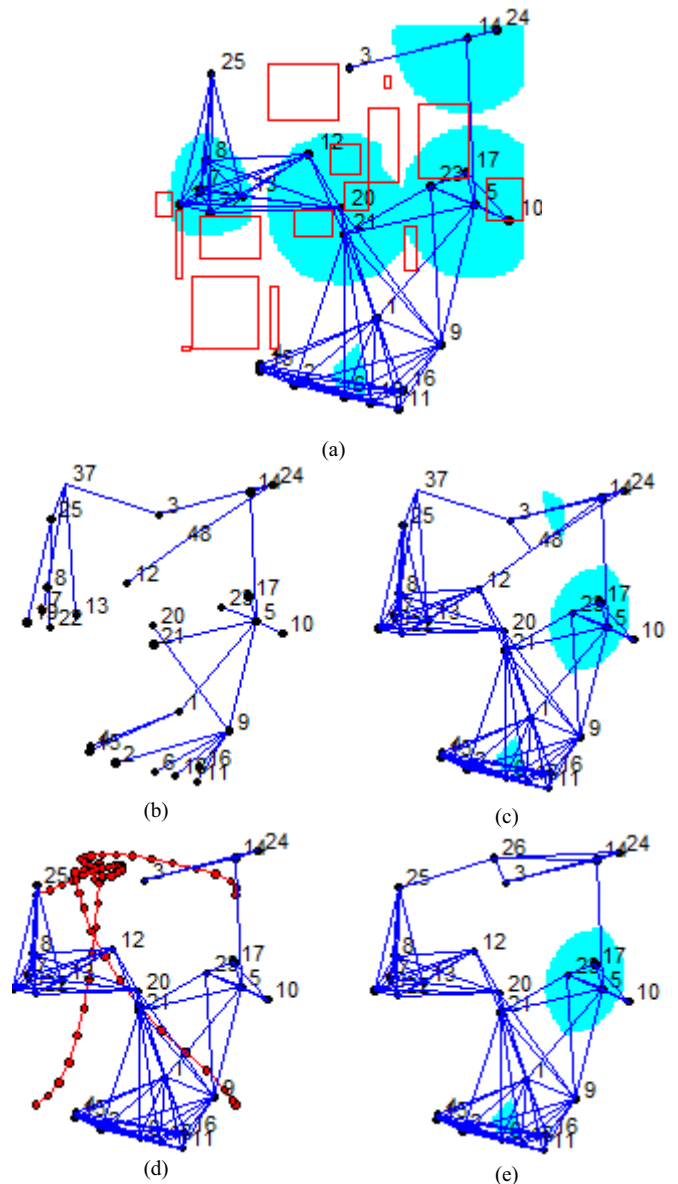


Figure 6. All-Terminal Example with ITU-R Model and Obstructions - Network B1 (25 Node) (a) Network B1 Showing the $V(N,E,r)$ prior to Augmentation (0.356). (b) ILP Optimized Outside Weighted Spanning Tree (Added Node 37, 48). (c) ILP Optimized, $V(N,E,r)$ of 0.090 (d) PSO Optimized (Added Node 26) (e) PSO Optimized, $V(N,E,r)$ of 0.082

Figure 6(a) shows significant vulnerabilities in 4 areas of the network with a vulnerability radius of 2.

Figures 6(b) and 6(c) show the ILP (spanning tree) approach with edge weighting (as opposed to center weighting). The vulnerability (0.090) after the addition of nodes 37 and 48 is shaded in Figure 6(c). As we can see, the new nodes eliminated most of the vulnerabilities. Typically, spanning trees on the edge produce better diversity from clusters of nodes in the center of the space. Addition diversity weighting added 4 nodes and removed all vulnerabilities as noted in Table III.

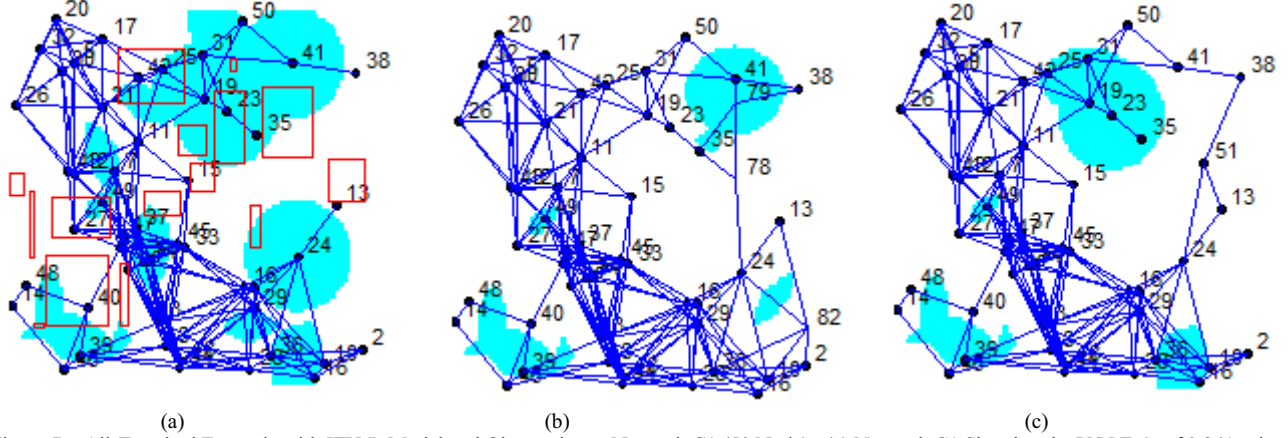


Figure 7. All-Terminal Example with ITU-R Model and Obstructions - Network C1 (50 Node). (a) Network C1 Showing the $V(N,E,r)$ of 0.341 prior to Augmentation. (b) ILP Optimized, $V(N,E,r)$ of 0.099 (Added Nodes 78, 82, 79). (c) Swarm Optimized, $V(N,E,r)$ of 0.147 (Added Node 51).

The PSO approach is shown in Figure 6(d) and 6(e). As shown in Figure 6(d), swarm movement did not find a good enough solution to stop based on the solution found (0.082). All 16 iterations were used before stopping. It is interesting to note that the PSO method found a “better” solution than the ILP approach with less nodes added.

C. Network C1 – All-Terminal Case

Figure 7 shows a 50 node example (Network C1) for the all-terminal case. In Figure 7(a), we see the connectivity generated by the ITU-R Pedestrian Model with random obstructions. The non-optimized network in Figure 7(a) shows significant vulnerabilities in the network and $V(N,E,r)$ of 0.341. A vulnerability radius of 1.5 was used.

Figure 7(b) shows the results of the ILP (spanning tree) approach with edge weighting and vulnerability of 0.099. The results of the PSO approach are shown in Figure 7(c). The vulnerability by adding node 51 is 0.147.

V. SUMMARY AND OBSERVATIONS

The goal of this work was to develop methods to identify and mitigate geographic vulnerabilities in networks. Two scenarios were considered. The point-to-point scenario looks at providing geographic diversity between two paths on a given demand. The all-terminal scenario seeks to develop network configurations to protect the connectivity of all nodes in the network from geographic impacts. Two approaches were used. The first is an ILP approach that used a geographic weighting scheme to encourage path 1 to locate in a different geographic area of the network than path 2 and spanning tree 1 (all-terminal) from spanning tree 2. The second was a PSO approach that incorporated $V(N,E,r)$ as the “non-linear” objective. This was used for both the point-to-point and all-terminal scenarios.

Since $V(N,E,r)$ is non-linear, the ILP approach cannot use that function directly. Thus the ILP method produces results that *approximate* minimizing $V(n,e,r)$. The weight parameters in the ILP approach are strongly topology dependant. A small increase in the diversity weight can cause several nodes to be added drastically reducing $V(n,e,r)$. More research needs to be

done to study the effect that the weight parameters have on solutions with different topologies.

The PSO approach adds one node at a time as it attempts to minimize $V(N,E,r)$ directly. This can produce optimal solutions. However, PSO is a heuristic and therefore it is possible to not locate the optimal solution. In our problem, generally we are looking for “good” solutions not necessarily “optimum” solutions to produce low or zero geographic vulnerability. Given that assumption, the PSO approach seems to produce good solutions in most cases.

The disadvantage of the PSO approach over the ILP approach is the complexity of the PSO approach. In this work, we solved all problems using Matlab [18] (with lp_solve [17] for ILP) installed on a Windows XP desktop computer. $V(N,E,r)$ in the worst case calculates a shortest path calculation (point-to-point) or Eigenvalue decomposition (all-terminal), a maximum of 2^N times, where N is the number of nodes. We are currently working on methods to reduce the complexity of the geographic evaluation function. In this work, the ILP approach calculation averaged approximately 1.0 seconds for the 15, 25, and 50 node problems. The PSO approach averaged 25.0 seconds for the 15 node problems, 172.0 seconds for the 25 node problems and 1500 seconds for the 50 node problems. The ILP approach has clear advantages in this regard.

There are several areas of promising research related to this work. First, we believe that an algorithm can be designed that optimizes the ILP weights to provide the best possible solution in a reasonable time frame. In addition, we plan to expand the ILP approach to include multiple grid configurations and different weighting schemes. The expansion of the PSO approach could include different numbers of particles, various acceleration constants, and different initial particle locations designs.

REFERENCES

- [1] K. Stephan, "We've Got to Talk: Emergency Communications and Engineering Ethics", *IEEE Technol. Soc. Mag.*, Vol. 26, pp. 42-48, Fall 2007.
- [2] Savannah River National Laboratory, "Urban Search and Rescue Technology Needs – Identification of Needs", Research Report prepared for the US Department of Justice, June 2004
- [3] M. Gardner, C. Beard, "Evaluating Geographic Vulnerabilities in Networks", *2011 IEEE Int. Communications. Quality and Reliability (CQR) Workshop*, May 10-12, 2011.
- [4] H. Gao, G. Barbier, R. Goolsby, "Harnessing the Crowdsourcing Power of Social Media for Disaster Relief", *IEEE Intell. Syst.*, pp 10-14, May/June 2011, IEEE Computer Society.
- [5] G. Clarke, "Megaquake Cuts Japan Phone Lines, Data Centers Successfully Fail Over", *The Register*, March 11, 2011, http://www.theregister.co.uk/2011/03/11/japan_quake_phones_out/
- [6] "Joplin Tornado Death Toll Jumps to 89", *The Wichita Eagle*, May 22, 2011, <http://www.kansas.com/2011/05/22/v-print/1859953/tornado-strikes-joplin-mo.html>
- [7] J. Zhou and C. Beard, "Balancing Competing Resource Allocation Demands in a Public Cellular Network that Supports Emergency Services," *IEEE J. Sel. Areas Commun., Special Issue on Mission Critical Networking*, vol. 28, no. 5, pp. 644-652, June 2010.
- [8] W. Dotson, J. Gobien, "A new Analysis Technique for Probabilistic Graphs", *IEEE Trans. Circuits and Systems*, Vol. CAS-26, No. 10, October 1977.
- [9] M. Fiedler, "Algebraic Connectivity of Graphs", *Czechoslovak Math Journal*, 23(98):298–305, 1973.
- [10] S. Neumayer, E. Modiano, "Network Reliability With Geographically Correlated Failures." *Proc. IEEE INFOCOM*, 2010, 1-9. IEEE
- [11] E. Cetinkaya, D. Broyles, A. Dandekar, S. Srinivasan, J. Sterbenz, "A Comprehensive Framework to Simulate Network Attacks and Challenges", *Proc. 2010 Int. Congr. Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp 538-544, 2010 IEEE.
- [12] R. Li, X. Wang, X. Jiang, "Network survivability against region failure," *Proc. 2011 Int. Conf. Signal Processing, Commun. and Comput. (ICSPCC)*, vol., no., pp.1-6, 14-16 Sept. 2011
- [13] Sen, A.; Murthy, S.; Banerjee, S.; , "Region-based connectivity - a new paradigm for design of fault-tolerant networks," *2009 Int. Conf. High Performance Switching and Routing (HSPR)*, vol., no., pp.1-7, 22-24 June 2009
- [14] M. Pioro, D. Medhi, *Routing, Flow, and Capacity Design in Communications and Computer Networks*, Elsevier Publishing, 2004.
- [15] A. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons Ltd, 2005
- [16] A. Molisch, *Wireless Communications*, John Wiley & Sons Ltd, 2005
- [17] M. Berkelaar, K. Eikland, P. Notebaert, *lp_solve Version 5.5.2.0*, 2004, <http://lpsolve.sourceforge.net/5.5/>
- [18] MATLAB version 7.5.0. Natick, Massachusetts: The MathWorks Inc., 2003

APPENDIX I

$V(N,E,r)$ is calculated using an algorithmic approach, which is described in detail in [3]. This approach is summarized here.

Notation

r	Threat radius
ISO	Matrix of feasible node combinations with a threat of radius r
F	List of feasible failure events
Q	FIFO Queue of network events to test

$E2$ Event of form $([x \ \overline{y}])$, where node x is active and node y is failed (for 2-Terminal method).
 Ea Event of form $([1,0,1])$ where 1 is not failed and 0 is failed (for all-terminal method).
 $G(V,E)$ Graph consisting of Nodes N and Edges E
 $G_E(V_E,E_E)$ Graph formed from $G(V,E)$ formed by Event E
 $L(G)$ Laplacian Matrix of G
 $\lambda_1, \lambda_2, \dots, \lambda_n$ Eigenvalues of $L(G)$

Calculating $V(N,E,r)$ for 2-Terminal Problem:

1. Use $G(N,E)$ and r to calculate ISO .
2. Set F to $[]$
3. Add $E2 = []$ network event to Q .
4. Loop till Q is empty
 - a. Pop event $E2$ off Q . Create $G_{E2}(V_{E2},E_{E2})$
 - b. Find shortest path in new G_{E2} using Breadth First Shortest algorithm (BFS). *success* if path found.
 - c. If (*success*), create new events reflecting the complement of new path not specified in $E2$.
 - i. Test new events for feasibility using ISO
 - ii. If (*feasible*) then add event to Q else discard
 - d. If (*not success*) add new failure mode to F
5. For each failure mode i in F
 - a. Find first node associated with failure mode i . Add threat radius around node as V_i area.
 - b. For remaining nodes n associated with failure mode i : Replace V_i with intersection of V_i and threat radius around node n . Repeat until no nodes remain in i .
6. Vulnerable area $V = V_1 \cup V_2 \cup \dots \cup V_F$
7. $V(N,E,r) = V / (\text{Total Network Area})$

Calculating $V(N,E,r)$ for the All-Terminal Case:

1. Use $G(N,E)$ and r to calculate ISO .
2. Set F to $[]$
3. Add $Ea = [1,1,1,\dots,1]$ network event to Q
4. Loop till Q is empty
 - a. Pop Ea off Q . Create $G_{Ea}(V_{Ea},E_{Ea})$
 - b. Calculate $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ of $L(G_E)$. If $\lambda_2 > 0$ then *success* (network connected).
 - c. If (*success*), create new events by incrementally failing each node remaining that is not part of Ea , add the new event to Q .
 - i. Test new events for feasibility using ISO
 - ii. If (*feasible*) then add event to Q else discard.
 - d. If (*failure*) store new failure event in F .
5. For each failure mode i in F
 - c. Find first node associated with failure mode i . Add threat radius around node as V_i area.
 - d. For remaining nodes n associated with failure mode i : Replace V_i with intersection of V_i and threat radius around node n . Repeat until no nodes remain in i .
6. Vulnerable area $V = V_1 \cup V_2 \cup \dots \cup V_F$
7. $V(N,E,r) = V / (\text{Total Network Area})$.