AUTOMATED RADIO NETWORK DESIGN USING ANT COLONY OPTIMIZATION

by

Jeffrey Allen Sharkey

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2008

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Radio networks can provide reliable communication for rural intelligent transportation systems (ITS). Engineers manually design these radio networks by selecting tower locations and equipment while meeting a series of constraints, such as coverage, bandwidth, maximum delay, and redundancy, all while minimizing network cost. As network size and constraints grow, the design process can quickly become overwhelming. Instead, we can automate the design process by modeling it as a generalized Steiner tree-star (GSTS) problem. Any minimum Steiner tree (MST) solution to the GSTS problem directly identifies the tower locations and equipment needed to build the network at an optimal cost.

Because the MST problem is known to be NP-hard, our research applies ant colony optimization (ACO) to quickly find near-optimal MST solutions. Using ACO also allows us to meet defined constraints while minimizing network cost. We verify that our approach finds near-optimal designs by comparing it against a 2-approximation algorithm in several different scenarios.

INTRODUCTION

Wired and radio links both provide communication for a variety of applications, from public cellular networks to private backhaul networks. Each link type has its own benefits and drawbacks. Radio links offer relatively inexpensive communication over long distances because they do not require leasing or installation of fiber or other wired infrastructure. However, in urban areas radio links can be less reliable because of interference and multipath fading [1]. In addition, radio links must be carefully tested for feasibility in light of environmental conditions, such as elevation and vegetation.

Each wired or radio link is defined with two endpoint nodes, and has an available bandwidth and expected delay associated with it. Multiple links can be combined in a chain to provide end-to-end communications. Nodes connected to two or more links are called relay nodes and may offer routing services between these links.

Typical wide-area networks (WANs) are designed using a combination of both wired and radio links [2] to meet a set of constraints while minimizing the cost of building the network. Finding these designs is generally known as the network design problem (NDP). The design process may consider thousands of possible relay nodes and links while looking for a minimal-cost subset that meets all constraints.

Figure 1 shows an example WAN built using a combination of wired and radio links. The links and relay nodes shown are only a small subset of those actually considered during the design process. Wired links are shown as solid edges, radio links as dashed edges, and the selected relay nodes as white vertexes.

As already mentioned, each WAN design must meet a set of constraints. These constraints can be expressed in terms of coverage, bandwidth, maximum delay, and redundancy:

- A coverage constraint describes a location or region to which the WAN must provide network connectivity. This could be a specific node, such as a stationary

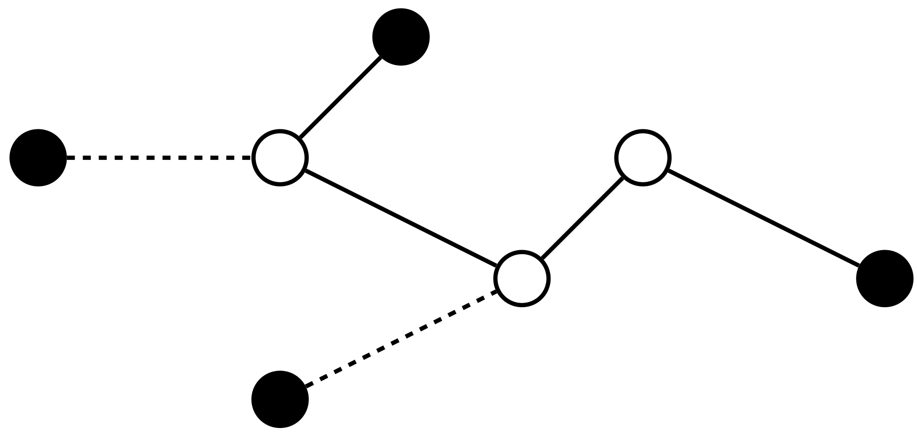


Figure 1. Example Wide-Area Network.

closed-circuit television (CCTV) camera, or a larger region, such as a one-mile segment of highway.

- A bandwidth (or capacity) constraint describes the minimum bandwidth required by specific nodes in the network. As network traffic is aggregated through various relay nodes, each link must have enough available bandwidth to satisfy all bandwidth constraints. This is also known as the capacitated network design problem (CNDP).
- A maximum delay constraint describes the maximum acceptable time for a single message to travel between two nodes. Delay constraints are important in Voice over IP (VoIP) deployments, where excessive delay is unacceptable [3].
- A redundancy constraint describes a pair of nodes that need to have redundant paths through the network. This ensures that the nodes can remain connected even if one of the paths fail. This is also known as the survivable network design problem (SNDP).

The general NDP allows connections to be relayed through any number of intermediate nodes. Some specific NDP problems can be simplified by assuming that only a static number of layers exist on the network. For example, most cellular network designs only required a two-layer design, where the first layer connects from mobile phones to the tower, and the second layer from the tower to the global backbone.

Traditionally, network engineers have manually found solutions to the NDP. However, as constraints grow, so does the complexity of the NDP. Evaluating each possible solution is infeasible, both manually by engineers and computationally.

Thus, the development of a metaheuristic approach to solving the NDP is important to quickly find minimal-cost solutions that meet all defined constraints. This thesis develops such a metaheuristic.

PREVIOUS WORK

Previous work has developed several automated approaches to solving the network design problem (NDP). Depending on the formulation, the NDP is either NP-complete [4] or NP-hard [5], and thus not optimally solvable in polynomial time.

In this section we examine existing approaches and find that none of them solves the NDP while meeting coverage, bandwidth, maximum delay, and redundancy constraints. We categorize each approach by the general strategy used.

Approximations

Several sources present integer linear programming (ILP) formulations to solve the network design problem (NDP). Since ILP problems cannot be solved in polynomial time, these sources also develop approximation algorithms. These approximations are formally proven to always produce NDP solutions within a bounded error ϵ of the optimal-cost NDP solution.

While the performance bounds of approximation algorithms are theoretically valuable, they can be unacceptable for real-world problems. For example, a 2-approximation NDP solution might cost twice as much as the actual optimal design. Because network cost is important, approximations without tight bounds are not directly useful.

[6] considers the capacitated survivable network design problem (CSNDP), which for each connectivity requirement (v_i, v_j, k) provides k edge-disjoint paths between v_i and v_j. They also handle unique bandwidth constraints for each edge, and develop a $2\min\{\mathcal{H}(f_{max}), q\}$-approximation algorithm where f_{max} is the maximum connectivity requirement, $\mathcal{H}$ is the harmonic series, and q is the number of distinct k constraints. Their approach does not consider maximum delay constraints, and the approximation bounds remain large.

[7] develops a linear programming representation for the survivable network design problem (SNDP). Similar to [6], connectivity requirements (v_i, v_j, k) are designed to provide k edge-disjoint paths between v_i and v_j. They then develop a $2\min\{\log R, p\}$-approximation algorithm, where R is the largest redundancy requirement k and p is the number of unique k constraints. Their approach does not consider bandwidth or maximum delay constraints, and the approximation bounds remain large.

[8] handles redundancy by creating a 2-connected sensor network where each sensor is covered by at least two relay nodes, and where two node-disjoint paths exist between each pair of relays. They develop an ILP formulation and a $D\log n$-approximation algorithm where D is the (2,1)-*Diameter* of the network and n is the number of sensor nodes. The (2,1)-*Diameter* is formally defined as the maximum length of all 2-connected paths between any two nodes (x, y) that does not include an arbitrary node z. [9] solves a similar problem while maximizing network lifetime based on battery life. Neither approach considers bandwidth or maximum delay constraints.

[10] designs a backhaul network by selecting a subset of existing WiMAX mesh nodes to be converted to base stations. They present a mixed-integer linear programming (MILP) formulation and show that its running time is unacceptable; several days for networks with more than 15 nodes. Instead, they develop a minimum cost flow (MCF) adaptation with a linear programming formulation, and use MCF solutions in an iterative process to guide the formation of a near-optimal network design. They compare their iterative approach against MILP solutions, and conclude that their approach finds designs very close to the optimal. While their approach considers bandwidth constraints, it does not consider redundancy or maximum delay constraints.

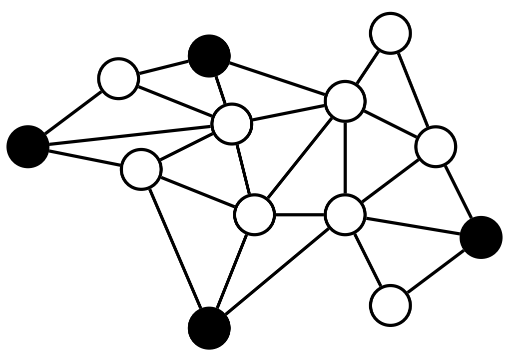
Steiner Graph Representation

Many approaches have modeled the network design problem as a Steiner graph $G = (V, E)$ with edge weights w_{ij} and a subset $S \subseteq V$ labeled as Steiner vertices. A Steiner tree-star graph is one variation which explicitly states that $\{(i, j) \mid i, j \notin$

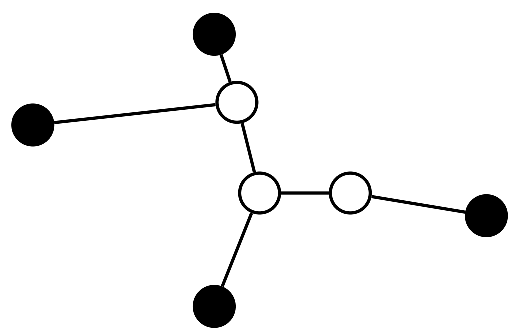
$S\} = \emptyset$. That is, all terminal, non-Steiner vertices are required to be leaf nodes. A generalized Steiner tree-star (GSTS) graph is another variation which has vertex weights w_i in addition to the edge weights. Given any Steiner graph, the minimum Steiner tree (MST) problem is to find a minimum-cost spanning tree that connects all terminal, non-Steiner nodes.

Steiner nodes are analogous to possible relay nodes, where a MST solution selects a subset of those nodes to build the network. By setting vertex and edge weights as real-world costs, the MST problem will minimize the cost of any network design.

Figure 2 *(a)* shows an example GSTS graph, where Steiner nodes are hollow (white) vertexes and terminal, non-Steiner nodes are solid (black) vertexes. In Figure 2 *(b)*, one possible MST solution to the given GSTS example is shown.



(a)



(b)

Figure 2. Example GSTS Graph and One MST Solution.

[11] develops an ILP formulation for solving the MST problem in Steiner graphs with $2|E|+(|V|-1)$ ILP variables and $4|E|+3|V|+|S|-4$ constraints. Their approach is a solid foundation to solving the MST problem, but they do not develop any approximations. In addition, their approach does not consider bandwidth, maximum delay, or redundancy constraints.

[5] develops a 5-approximation algorithm to solve MST in GSTS graphs. They adapt existing approximations for the uncapacitated facility location (UFL) problem and adapt them to solve the MST problem. Their approach does not consider bandwidth, maximum delay, or redundancy constraints, and the approximation bounds remain large.

[12] develops a 2.5-approximation algorithm for solving the MST problem in Steiner graphs. They apply the approximation to solving the NDP of selecting relay nodes in a wireless sensor network. Their approach allows for bounded edge lengths to maximize network lifetime. However, their approach does not consider bandwidth, maximum delay, or redundancy constraints, and the approximation bounds remain large.

[13] solves a variation of the GSTS where connectivity requirements are given as node pairs (v_i, v_j, r) instead of the entire MST problem. Similar to [6], each requirement says that r edge-disjoint paths must exist between v_i and v_j. They develop a $2\lceil\log_2(r_{max}+1)\rceil$-approximation where r_{max} is the highest r constraint. Their approach does not consider bandwidth or maximum delay constraints, and the approximation bounds remain large.

[14] develops a $2(1-\frac{1}{l})$-approximation to solving the MST in Steiner graphs where l is the number of leaves in the optimal tree. In GSTS graphs, the number of leaves is exactly the number of terminal nodes $|\{v \mid v \notin S\}|$. Their approach does not consider bandwidth, maximum delay, or redundancy constraints, and approximation bounds remain large.

[15] develops a 2-approximation for the MST problem in GSTS graphs with the assumption that all edge and vertex costs are unique. We compare our metaheuristic against this approximation because it has a low bound while still solving a Steiner representation of the NDP. Their approach does not consider bandwidth, maximum delay, or redundancy constraints, and the approximation bounds remain too large to provide a direct reasonable solution.

[16] develops a 1.598-approximation algorithm by assuming that the Steiner graph can be considered complete with new edge weights assigned as the shortest path length. They use a previous approach [17] in an iterative framework to reach their bounds. Their assumptions about the behavior of edge weights only hold for geometric cases, and thus this approximation cannot be used to solve the NDP.

[18] develops a $1 + \frac{ln3}{2}$-approximation algorithm similar to [16]. Their approach assumes that edge weights are metric (the triangle inequality holds) which only holds for geometric cases, and thus this approximation cannot be used to solve the NDP.

Metaheuristics

Metaheuristics are another method of solving NP-complete and NP-hard problems in polynomial time. They are artificially intelligent algorithms designed to search for good NDP solutions while meeting constraints.

Genetic Algorithms

Genetic algorithms (GAs) start with a population of solutions, and apply mutation and crossover operations to create new solutions. A fitness function is then used to select a subset of these new solutions to be included in the next population. Each iteration through the algorithm is called a generation.

Several sources have applied GAs to the NDP of selecting tower locations for cellular networks which are modeled as two-layer networks with the mobile handset connecting to a local cell and each cell connecting directly to a global backbone. Redundancy is important in cellular networks so that calls are not dropped as mobile handsets pass between cells. While low delay is important in cellular networks, maximum delay constraints are not relevant because there are only two layers in the network.

[19] models the NDP as a minimum dominating set (MDS) problem and solves it using multiple GA instances running in parallel. They consider bandwidth as the

number of calls that each tower needs to handle, but do not consider redundancy. Their coverage requirement simply tries to maximize the ground area covered by towers, and does not handle explicit coverage constraints.

[4] develops a GA to design radio networks using microcells that only cover small areas. They assume that all bandwidth constraints and node costs are uniform, and they do not consider redundancy constraints. Similar to [19], they focus simply on maximizing coverage area.

[20] also addresses the design problem using a GA. They handle coverage constraints as reception test points (RTP) which must have signal, service test points (STP) which must have an acceptable signal-to-noise ratio, and traffic test points (TTP) which also serve as bandwidth constraints. They also handle redundancy by requiring overlap between each cell to facilitate mobile hand-over.

[21] considers the case where cellular towers are already placed. Their approach designs the wired network from each existing cellular tower back to base station controllers and mobile switching centers. They also consider the bandwidth requirements of each cell. Their approach does not consider redundancy or maximum delay constraints.

[22] develops a method of finding the Pareto front between cellular coverage and cost over a region. In the process, they develop a GA to place cellular towers using the Hata propagation model to validate radio paths. Their approach does not consider bandwidth, maximum delay, or redundancy constraints.

[23] considers the design of wireless sensor networks. They develop a multiobjective GA where the competing objectives are cost and network lifetime based on battery life, similar to the problem described in [9].

Simulated Annealing

Simulated annealing (SA) is a local search technique that uses a single solution to derive neighboring solutions, and then probabilistically move toward neighbors with lower costs. As a SA algorithm progresses, it "cools" which reduces the probability of selecting a non-optimal neighbor.

[24] develops a SA approach to designing multi-layered telecommunication networks using leased wired infrastructure. They consider bandwidth demands of clients and focus on link pricing based on varying capacity. They also consider redundancy requirements and add additional links between add drop multiplexers (ADM) as needed. They compare SA results against against a LP relaxation and find 33% better results.

[25] develops a SA approach that also finds optimal antenna parameters such as power, azimuth, and tilt. Their approach focuses on cellular networks and thus designs only two-layer networks. They address minimizing interference between multiple sites, and verify that at least one coverage point in each cell has hand-over coverage provided by four other cells. They consider bandwidth requirements as the number of simultaneous calls that each cell must handle.

Tabu Search

Tabu search (TS) is a local search technique similar to simulate annealing. It uses a single solution to derive neighboring solutions, and then probabilistically moves toward neighbors with lower costs. The algorithm maintains a "tabu" list of all previously visited solutions to prevent looping and avoid local optima.

[26] develops an ILP that solves the two-layer capacitated network design problem (CNDP). They develop an approximation based on algorithms for solving the multiple-choice multidimensional knapsack problem (MCMKP). They use a greedy algorithm to form an initial solution, and then apply TS to find a more optimal solution. While their approach considers bandwidth constraints, it does not consider maximum delay or redundancy constraints.

[27] models the NDP as a Steiner tree-star graph and solves the MST problem using TS. In each iteration they swap Steiner nodes to create new solutions from an existing solution list. The tabu list prevents them from selecting swaps that violate

any solution constraints. Also, after a swap they prevent its reversal for a given number of generations. They compare their results against a local search heuristic, and confirm that their TS approach finds the optimal solution for several test cases. Their approach does not consider bandwidth, maximum delay, or redundancy constraints.

[28] applies TS to find an optimal multicast routing tree while meeting quality-of-service (QoS) constraints. They consider both bandwidth and delay constraints, and show that their TS approach performs better than several other heuristics. However, their approach does not consider redundancy constraints.

Ant Colony Optimization

Ant colony optimization (ACO) models the behavior of real-world ants as they search for food. As the ants search, they leave a trail of pheromone behind to help guide future ants towards optimal paths to food. As time progresses, the pheromone on non-optimal paths evaporates while the pheromone on near-optimal paths is reinforced. Figure 3 illustrates how, after time, pheromone is concentrated (darker) on a short path around the obstacle, thus influencing more ants to follow that path.

In ACO, ants construct solutions by traversing a graph using a local heuristic and existing pheromones as a guide. Several ants work together to construct a population of solutions, called a generation. At the end of each generation, the best solution is used to reinforce certain pheromones while evaporating others. This process is repeated for a set number of generations, or until the pheromones converge.

[29] develops an ACO approach for solving the MST problem on traditional Steiner graphs. They place ants on each terminal node in the Steiner graph. As each ant moves across the graph, the local heuristic tries to merge towards any nearby ant trails. Once all ants have merged, the ACO solution is an MST solution. However, this approach does not consider bandwidth, maximum delay, or redundancy constraints.

[30] develops an ACO approach to solving the MST problem on rectilinear Steiner graphs. Rectilinear Steiner graphs are defined in euclidean space with the constraint

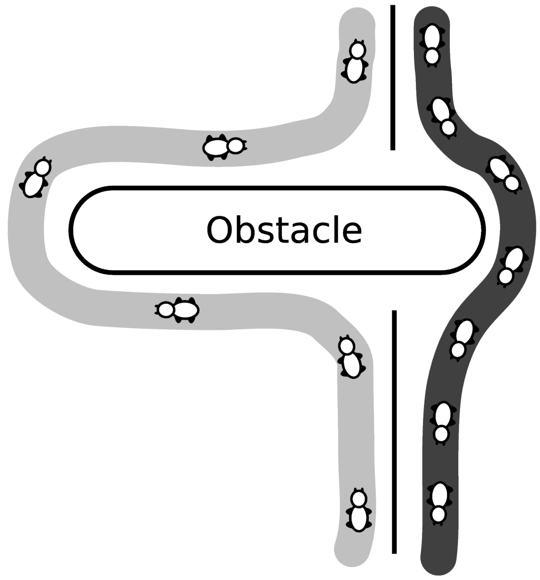


Figure 3. Pheromones on Ant Trails.

that all edges must be strictly horizontal or vertical. Solutions to this problem are important to finding optimal wire routing in circuit design. However, because of the euclidean constraints this approach cannot be used to solve the NDP.

Limitations

After a survey of the NDP, we found that several approaches have been developed to find solutions in polynomial time. However, some approaches make assumptions about circular or uniform radio coverage [12] [8] [19] [23] [4] [31], which do not hold true over non-trivial distances in rural, mountainous areas.

Other approaches consider only two-layer network designs [26] [25] [19] [20] [4], which can be infeasible in rural areas where the ability to relay through several layers can be critical to optimal network design.

Only a few algorithms consider bandwidth and redundancy constraints [8] [24] [6], but even then only in a limited way. In addition, only one algorithm considers delay constraints [28].

Approximation algorithms exist for the MST problem, but the best applicable approach is a 2-approximation [15] that does not consider bandwidth, delay, or redundancy constraints.

METHODOLOGY

Propagation Model

There are two general methods of determining the feasibility of a given radio link. One method is to assign each node a radius of communication based on radio power and antenna gain [2] [12]. This results in a circular coverage area with the node at the center. A link is considered possible when each node is within the coverage area of the opposite node. While this method is simple, it does not accurately model a real-world environment where terrain can drastically change the coverage area of a given node.

Another method is to examine the terrain along the path between the two nodes. Several algorithms follow this method, and many radio engineers agree that the Irregular Terrain Model (ITM), also called the Longley-Rice Model, is one of the most accurate [32]. ITM is based on solid propagation theory and has been well-tested in various terrain and environmental conditions.

The ITM estimates the expected power loss ITM_{loss} of a radio signal along a given path. At the transmitter, the equivalent isotropically radiated power (EIRP) is given in dBm by Equation 3.1, where p is the transmitter power in watts, and a_{tx} is the transmitter antenna gain in dBi.

$$\text{EIRP} = 10 \log \frac{p}{1mW} + a_{tx} \qquad (3.1)$$

Given a transmitter EIRP and path loss ITM_{loss}, the receiving radio has a sensitivity r given in dBm. This sensitivity indicates the signal strength required to successfully decode a message from the transmitter. Some radios have multiple bandwidths depending on the received signal strength, as shown later in Table 1. Typically a stronger signal will allow for higher bandwidths. Equation 3.2 shows the relationship that needs to hold for a signal to be decoded at a given receiver with sensitivity r and antenna gain a_{rx}.

$$\text{EIRP} - \text{ITM}_{loss} + a_{rx} > r \tag{3.2}$$

To best model real-world conditions, our approach uses the ITM to analyze radio link feasibility. We use high-resolution elevation data to maintain accuracy in extreme terrain conditions [33]. Because the ITM does not consider foliage, we augment it with Weissberger's Modified Exponential Decay Model to estimate path loss caused by any trees in the radio line-of-sight path [34].

Generalized Steiner Tree-Star Adaptation

Similar to [5], we model the network design problem as a generalized Steiner tree-star (GSTS) graph. A GSTS graph is a graph $G = (V, E)$ with both edge weights w_{ij} and vertex weights w_i. A subset $S \subseteq V$ are called Steiner vertices. The GSTS graph is a natural representation for the NDP, where coverage constraints are terminal nodes, all considered relay nodes are Steiner nodes, and all considered links are edges. Any minimum Steiner tree (MST) solution to a GSTS graph directly identifies the relay nodes and links needed to build the network at an optimal cost.

In our construction, terminal nodes represent all coverage requirements. An additional terminal node is added to represent a global backbone. We make the assumption that coverage requirements describe which locations must be connected to this global backbone. Steiner nodes are inserted for each location where we consider placing a relay node. Methods for selecting possible relay node locations are described later. One possible method is to discretize an entire area to obtain a grid of possible relay nodes.

All nodes and edges are assigned costs, each edge has an available bandwidth and expected delay assigned. All terminal nodes have both a required bandwidth and maximum delay assigned.

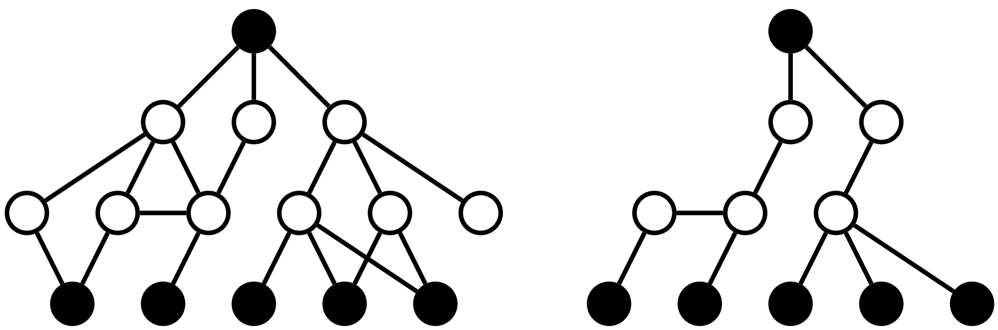
We formally construct the nodes in the GSTS graph as follows:

- We place a single terminal node at the top of the graph to represent the backbone of the network. It is assigned a zero cost, a zero required bandwidth, and an infinite maximum delay.
- For each coverage requirement, we place terminal nodes at the bottom of the graph. Each node is assigned a zero cost, a required bandwidth, and a maximum delay as specified by any constraints.
- We insert Steiner nodes for each location where we consider placing a relay node. Each relay node is assigned the cost of preparing that site for use as a relay. This cost might include baseline equipment or construction of a physical tower. Each relay node is also assigned a zero required bandwidth and an infinite maximum delay.

Then we add edges to the GSTS graph as follows:

- We add edges for any wired links being considered. Each edge is assigned the cost of installing and commissioning any required equipment. An available bandwidth and expected delay are also assigned based on equipment specifications.
- Similarly, for each radio link being considered, we test its feasibility using the ITM algorithm. If the link is possible with a given set of radio equipment, we create an edge between the two nodes. Each edge is assigned the cost of the actual radio equipment needed at both ends of the link. An available bandwidth and expected delay are also assigned based on equipment specifications.

Figure 4 *(a)* shows an example GSTS graph following the construction outlined above. Steiner nodes are hollow (white), terminal nodes are solid (black), and considered links are solid edges. Bandwidth and delay constraints are not show, and costs are assumed to be uniform. In Figure 4 *(b)*, one possible MST solution to the GSTS example is shown.



(a) *(b)*

Figure 4. Example GSTS Construction Graph and MST Solution.

Additionally, there are some special cases not handled by the above construction:

- When a coverage requirement is defined as a region, we discretize that region into a set of individual terminal nodes. Any maximum delay constraint is assigned, in full, to each terminal node. Regions cannot have bandwidth constraints.
- Some wired links may have recurring costs such as line leasing. To consider this in the network cost, estimate the lifetime of the network design and total any recurring costs over this lifetime. These total costs can then be applied to nodes and edges where needed. Other considerations, such as equipment depreciation and variable costs over time, are beyond the scope of this thesis.
- When locations are already developed, or equipment already exists, the assigned cost can be reduced to reflect this. For example, an existing radio link would be assigned zero cost while still having an associated available bandwidth and expected delay.
- Some relay nodes may use a single omnidirectional radio to connect with several other nodes. By moving the cost of the omnidirectional radio to the Steiner node

instead of leaving it on each edge, we can ensure that the overall network cost is calculated correctly.

While this construction is similar to previous approaches, it is novel in the way it distributes real-world costs. The construction also allows for connections to be relayed through any number of relay nodes before reaching the backbone.

Given the construction above, any solution to the MST problem is therefore a network design meeting all coverage constraints while minimizing cost. Note that the MST problem does not directly consider bandwidth, maximum delay, or redundancy constraints. Previous work has developed a 2-approximation algorithm for solving this MST problem [15]. We next formulate a metaheuristic approach to solving this MST problem while adding additional functionality to satisfy bandwidth, delay, and redundancy constraints.

Metaheuristic Formulation

Solving the minimum Steiner tree (MST) problem on GSTS graphs has been shown to be NP-hard [5]. To solve the problem in polynomial time, we apply an artificial intelligence metaheuristic called ant colony optimization (ACO) [35]. Specifically, we use the the $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) variation of ACO.

Following the approach described in [35] and [36], all pheromones are initialized to τ_{init} and we construct a population of p MST solutions. Each solution $S = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$, is constructed by a unique group of ants. Each population of solutions is called a single generation. At the end of each generation, we use the lowest-cost solution found so far to update the pheromones. We repeat this process for g generations, or until pheromones converge. The final output is the lowest-cost MST solution found over all generations.

For the construction of each solution, one ant will be placed at each terminal node at the bottom of the graph. Each ant is assigned the bandwidth required by and the maximum delay constraint of the terminal node it represents. We randomly select one of those ants to completely traverse the GSTS graph until it reaches the global backbone terminal node. When this ant is finished, we randomly select the next ant until all ants all reach the backbone. This random selection of ants is necessary to counter bias introduced by our serial construction method.

When an ant is selecting the next edge to follow, it considers the lowest cost that each path offers for reaching the global backbone, in addition to any pheromone left by previous ants. When at node i, the probability of following an edge to node j is given by Equation 3.3, where τ_j is the pheromone value for node j, and η_{ij} is the local heuristic defined later.

$$P(j|i) = \frac{\tau_j^{\alpha}\eta_{ij}^{\beta}}{\sum_{(i,k)\in E}\tau_k^{\alpha}\eta_{ik}^{\beta}} \tag{3.3}$$

We only consider edges in E that both have enough available bandwidth to support our bandwidth requirement and have an expected delay low enough to not violate our maximum delay requirement. When we follow an edge, we subtract our required bandwidth from that edge's available bandwidth. This ensures that future ants will not overload an edge's bandwidth. In addition, when we follow an edge we subtract its expected delay from our maximum delay requirement. This ensures that data from our terminal node will be able to reach the global backbone within the overall maximum delay required.

The probability function $P(j|i)$ mixes the pheromone and local heuristic values using α and β, respectively. These mixing variables control how much we depend on previous experience versus heuristic knowledge when constructing each solution. In our results we test various β values and compare their performance.

The local heuristic η_{ij} is dynamic and is computed as shown in Equation 3.4, where s_{ij} is the lowest-cost path from node i to j, node 0 being the backbone terminal node, V' being the set of vertices already selected by other ants, and ϵ being an arbitrarily small constant.

$$\eta_{ij} = \begin{cases} s_{j0} & \text{if } i \in V' \text{ and } j \in V' \\ 2 \times s_{j0} & \text{if } i \in V' \text{ and } j \notin V' \\ min\{s_{j0} - \epsilon, min_{k \in V'} s_{jk}\} & \text{if } i \notin V' \end{cases} \qquad (3.4)$$

Similar to [29], this heuristic tries to merge the ant with any existing nearby ant trails, otherwise it moves towards the backbone node. Once the ant has merged with another trail ($i \in V'$), it is encouraged to move along that existing trail toward the backbone ($j \in V'$). If all nodes $j \in V'$ cannot be followed because of some bandwidth or delay constraint, then we consider nodes outside of existing trails ($j \notin V'$).

To illustrate this local heuristic, we look at examples of each case described above. Consider the GSTS graph in Figure 5 *(a)*, where the ant from t_3 has already reached the terminal node t_0. We assume that edge and node costs are uniform for this example. Considering the ant from t_1 currently at node a, the third clause of the local heuristic will favor moving to b or d because it will merge with an existing path. If the ant moves to either b or d, the first clause of the local heuristic will prefer following the existing trail to t_0. Any edge choices leading away from the existing trail are discouraged by the second clause.

If the GSTS graph were modified as shown in Figure 5 *(b)*, the third clause of the local heuristic would favor moving directly to t_0 because all edge costs are equal. The small constant ϵ ensures that the local heuristic will slightly prefer paths towards the global backbone when a tie exists. If the cost of (a, t_0) was high compared to (a, b), or (a, d), the local heuristic would instead favor merging with b or d, even though a direct edge to t_0 is available.

These examples illustrate how the local heuristic influences ants to merge trails as they build solutions, thus reducing solution costs.

As the ant moves through the graph, it maintains a tabu list of edges it has already visited. While this prevents looping, it can also result in dead-end solutions that are not completely solved. When we find a dead-end solution, we reset the solution $S = (\emptyset, \emptyset)$, place all ants back on their respective terminal nodes, and build the solution again.

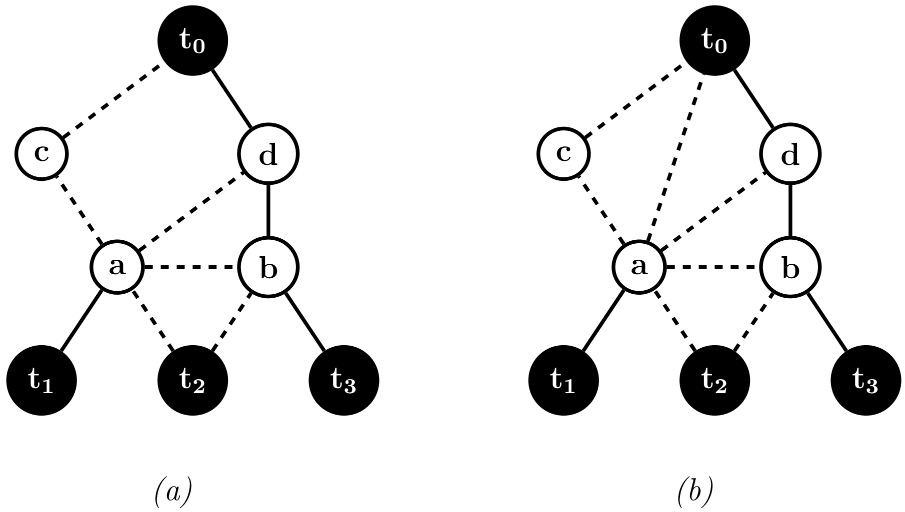


Figure 5. Example of ACO In Progress on a GSTS Graph.

Redundancy constraints can be defined as requiring n unique edge-disjoint paths with a r-edge relaxation. For each additional unique path required, we place another ant on the terminal node. All ants from a given terminal node use a shared tabu list, which prevents them from sharing any common edges as they construct paths to the global backbone. The r-edge relaxation allows ants to ignore this tabu list when they are less than r edges from reaching the global backbone. Without this relaxation, there may be no possible solution with n completely unique paths to the backbone.

After generating a population of p solutions, we update the pheromones across the graph as shown in Equations 3.5 and 3.6 using the lowest-cost solution $S^* = (V^*, E^*)$ found so far, where τ_{i_θ} is the pheromone for vertex i at generation θ, and ρ is a constant pheromone evaporation rate. This is based on the pheromone update process for $\mathcal{MM}$AS [36] [37].

$$\tau'_{i_\theta} = \begin{cases} \tau_{i_\theta} + 1/S^*_{cost} & \text{if } i \in V^* \\ (1-\rho)\tau_{i_\theta} & \text{if } i \notin V^* \end{cases} \tag{3.5}$$

$$\tau_{i_{\theta+1}} = max\{\tau_{min}, min\{\tau_{max}, \tau'_{i_\theta}\}\} \tag{3.6}$$

This update rule increases the pheromone of any nodes included in the solution S^*, while exponentially decaying the pheromone of all other nodes. We place pheromones on nodes instead of edges because solutions do not depend on construction order. Finally, we enforce a pheromone range $[\tau_{min}, \tau_{max}]$ before storing the new pheromone value.

As mentioned above, we repeat the population process until pheromones converge, or for a constant g generations. It is important to note that pheromone convergence results in a convergence in value. That is, solutions generated by a set of pheromones will be similar in cost, but may be very different in construction. We formally define pheromone convergence below, where ϵ is a small constant.

$$\{\tau_i \leq \tau_{min} + \epsilon \text{ or } \tau_i \geq \tau_{max} - \epsilon \mid \forall\, i \in V\} \tag{3.7}$$

That is, when all pheromones have moved within ϵ of the maximum or minimum value.

RESULTS

We tested our metaheuristic approach to solving the network design problem (NDP) by examining several real-world scenarios and comparing our solutions to a 2-approximation algorithm [15]. For each testing scenario we applied the above GSTS construction and ACO algorithm to find near-optimal NDP solutions. Because the 2-approximation algorithm requires unique edge and vertex costs [15], we randomly perturbed the assigned GSTS costs to have a 1% variation.

For testing we implemented both our ACO approach and the 2-approximation algorithm in C++ on a dual-processor Intel Xeon E5345 computer (eight 2.33GHz cores in total) with 16GB of RAM. We used Floyd's algorithm to generate a s_{ij} lookup table, and used modified standard template library (STL) data structures for storing the GSTS graph and solutions. We used the ITM propagation algorithm with 10-meter digital elevation model (DEM) data from the United States Geological Survey (USGS).

Variables

For our GSTS construction, we assigned real-world costs, available bandwidth, and expected delay with the guidance of a professional radio engineer. For equipment we assumed the use of typical 900MHz and 5.8GHz radios as shown in Table 1. As mentioned earlier, the expected bandwidth of a radio link is based on the sensitivity needed to receive a transmitted signal after considering ITM-estimated path loss.

The 900MHz system we selected operates in the license-free 902-928MHz ISM (industrial, scientific and medical) band at distances up to 40km. The 5.8GHz system we selected operates in the 5.725-5.875GHz ISM band at distances up to 80km.

Node variables, including bandwidth and maximum delay constraints, are shown in the first section of Table 2. Edge variables, including available bandwidth and

Table 1. Radio Sensitivity and Bandwidth.

Radio	Frequency	Power	Antenna	Sensitivity	Bandwidth
GE MDS Mercury 900	900MHz	1 watt	6dBi	-95dBm	500Kbps
				-92dBm	2.4Mbps
				-86dBm	4.8Mbps
				-77dBm	7.2Mbps
GE MDS Intrepid	5.8GHz	0.2 watts	13dBi	-84dBm	4.2Mbps
				-81dBm	6.5Mbps
				-74dBm	13.6Mbps
				-68dBm	18.3Mbps

expected delay, are shown in the section section of Table 2. In our construction we select only one receiver sensitivity value for each category of radio links as shown in the second section of Table 2.

Table 2. Assigned GSTS Variables.

Type	Frequency	Cost	Bandwidth	Delay
Terminal node		$0	varies	varies
Roadside relay node		$2,000	0	0
Existing backbone relay node		$10,000	0	0
New backbone relay node		$50,000	0	0
Terminal to Roadside relay	900MHz	$0	500Kbps	10ms
Roadside relay to Roadside relay	900MHz	$1,000	2.4Mbps	10ms
Roadside relay to Backbone relay	5.8GHz	$5,000	6.5Mbps	20ms
Backbone relay to Global backbone	wired	$0	10Mbps	30ms

For our ACO algorithm we set the variables α, β, τ_{min}, τ_{max}, and ρ as follows. Based on the approach in [37], our ACO algorithm uses $\alpha = 1$, $\tau_{min} = 2/\text{estimate-opt}$, $\tau_{max} = 5$, and $\rho = 0.95$, where estimate-opt is the solution cost found by the 2-approximation algorithm. Based on our analysis presented later, we set $\beta = 30$. For pheromone convergence testing, we set $\epsilon = 0.01$.

Testing Scenarios

We selected several segments of real-world highways to use as testing scenarios. Using these scenarios we solved two different NDP variations: one where we relayed a connection along the highway between two points, and one where we covered the entire roadway.

For our scenarios we selected segments along US-299, US-3, US-199, US-191, and US-101:

- US-299 runs between Arcata and Redding in northern California. It winds through mountainous terrain in a rural area, and existing cellular coverage is sparse. For the US-299 (a) scenario we considered one existing and one new backbone relay tower. For the US-299 (b) scenario we considered two new backbone relay towers.
- US-3 runs between Douglas City and Hayfork in northern California. It winds sharply through mountainous terrain in a rural area, and existing cellular coverage is sparse. For this scenario we considered two new backbone relay towers.
- US-199 runs between Crescent City, CA and Grants Pass, OR through the Smith River National Recreation Area. It winds sharply through heavily mountainous terrain in a very rural area, and existing cellular coverage is very sparse. For the US-199 (a) scenario we considered three existing backbone relay towers. For the US-199 (b) scenario we considered one existing backbone relay tower. Figure 6 illustrates the terrain along US-199.
- US-191 runs between Bozeman, MT and West Yellowstone, MT through the Gallatin River Canyon. It winds through mountainous terrain in a rural area, and existing cellular coverage is sparse. For this scenario we considered three new backbone relay towers. Figure 7 illustrates the terrain along US-191.
- US-101 runs along much of the California coastline. We considered an area around Ukiah in northern California that is relatively flat with straight road. For this scenario we considered one existing backbone relay tower.

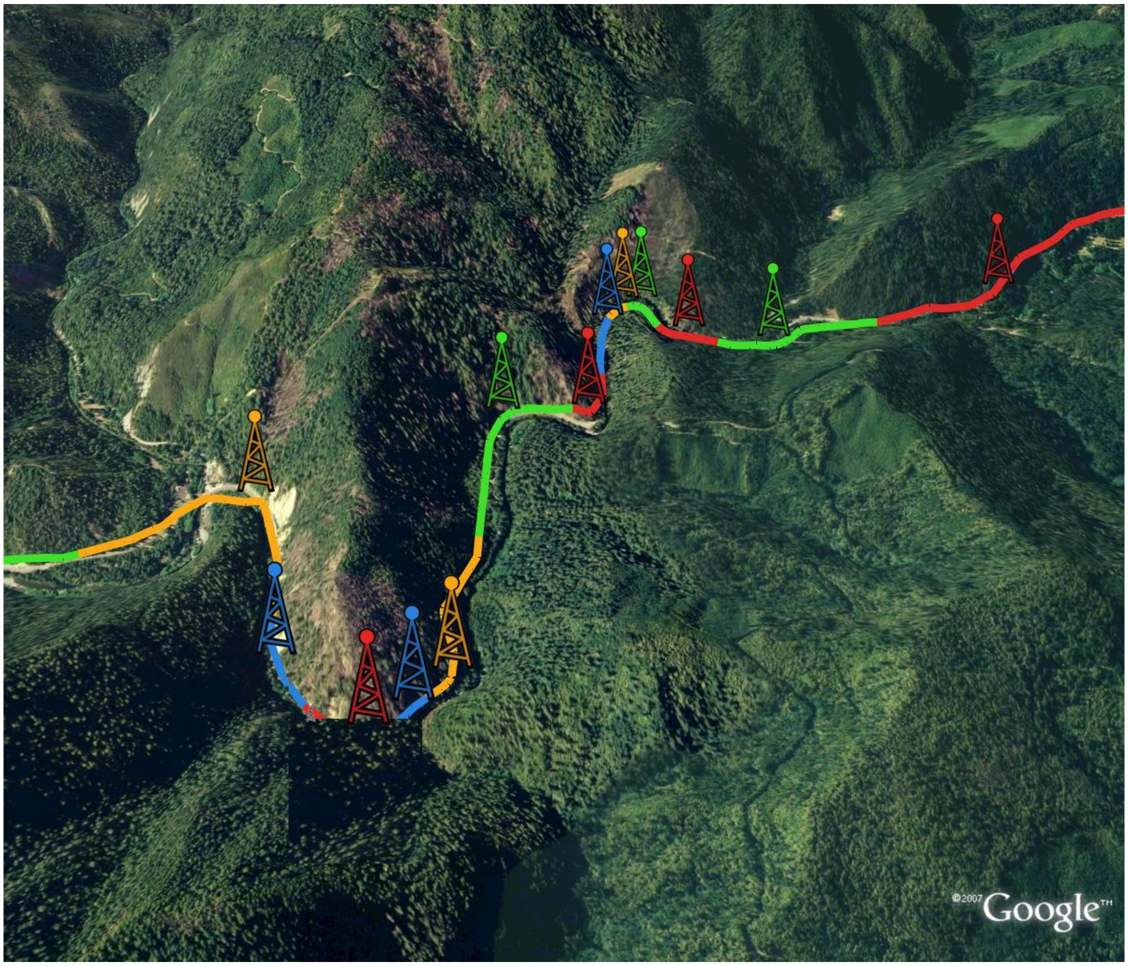


Figure 6. Partial ACO Solution for US-199 Scenario.

In our construction we differentiated between three types of relay nodes: roadside relay nodes, existing backbone relay nodes, and new backbone relay nodes. Roadside relay nodes can be placed anywhere along the side of a road. We consider the center of the road when placing relay nodes. Because the distance from the road center to either shoulder is about 10 meters, relay towers can be placed on either side of the road with minimal effect on propagation. However, with this assumption manual verification by an engineer should take place before actual network deployment.

Existing backbone relay nodes are tower sites gathered from publicly available FCC data. Usually these towers have several radios already installed, and additional

Figure 7. Partial ACO Solution for US-191 Scenario.

radios can be added. New backbone relay nodes are considered in areas without existing backbone relay nodes. We select sites on nearby mountain ridges that appear to have back-road access.

Different costs are assigned to each of these relay node types as listed in Table 2. We assume that all backbone relay nodes have direct global backbone connectivity.

Relay Along Roadway

First we considered the NDP of relaying a connection along a roadway between two points x and y. These two points are assumed to be at the roadside. For our

GSTS graph, we inserted both x and y as terminal nodes, but we consider x to be our global backbone terminal node. Then we discretized the roadway between the two points in 25-meter steps. These discretized points became the candidate relay locations and were inserted into the GSTS graph as Steiner nodes. All terminal and Steiner nodes were assigned costs as specified in Table 2. No bandwidth, maximum delay, or redundancy constraints were specified.

Then we used the ITM algorithm to test each pair of points in the GSTS graph, inserting edges where communication was possible based on the radio parameters given in Table 1. All edges were assigned costs, available bandwidth, and expected delay as specified in Table 2. Finally, we ran our ACO algorithm on this graph for 16 generations of population 64 each. For this specific NDP construction, the optimal solution can be found in polynomial time using Dijkstra's shortest-path algorithm. We tested this NDP construction to show show that our ACO algorithm quickly approaches known optimal solutions.

We solved this NDP construction for scenarios US-191 and US-199 (b). We first found the optimal NDP solution using Dijkstra's algorithm, and then ran our ACO algorithm. A summary of these two scenarios and their results are listed in Table 3.

Table 3. Relay Along Roadway Scenarios.

	US-191	US-199
Segment length	48km	43.3km
Vertices	1921	1732
Edges	144,732	52,410
Dijkstra cost	$83,014	$44,012
ACO cost	$83,028	$44,015
ACO generations	4	10

For the two scenarios tested ACO found solutions very near the known optimal, within $14 for US-191 and $3 for US-199 (b). These small cost differences compared to Table 2 are due to the perturbed GSTS costs. On average, ACO converged to its solution after 7 generations.

These results show that ACO can quickly find near-optimal solutions to this variation of the NDP.

Covering Roadway

The second NDP variation we considered was covering an entire segment of roadway so that a connection to the backbone could be made at any point along the roadway. This scenario is important for many applications, such as Vehicle Infrastructure Integration (VII) and roadside assistance.

For our GSTS graph, we inserted Steiner nodes for any considered backbone relay nodes. Then we discretized the roadway segment in 25-meter steps. For each discretized point along the road, we inserted both a terminal and a Steiner node into the GSTS graph: the terminal node representing the need to cover that point of roadway, and the Steiner node representing that we are considering placing a roadside relay at that point. All terminal and Steiner nodes were assigned costs as specified in Table 2. No bandwidth, maximum delay, or redundancy constraints were specified.

Then we used the ITM algorithm to test each pair of points in the GSTS graph, inserting edges where communication was possible based on the radio parameters given in Table 1. All edges were assigned costs, available bandwidth, and expected delay as specified in Table 2. When running the ITM propagation tests we did not consider the mobile case where nodes move along the highway.

We solved this NDP construction for scenarios US-299 (a), US-299 (b), US-3, US-199 (b), and US-191. We first ran the 2-approximation algorithm, and then ran our ACO algorithm for 16 generations of population 64 each. A summary of all tested scenarios and their results are listed in Table 4. The first section shows statistics including the road length considered and the number of vertices and edges in the constructed GSTS graph. The second section shows the running time in seconds taken for each portion of our approach. Both the ITM algorithm and Floyd's algorithm must be run to construct the GSTS graph, and are therefore required overhead for both the 2-approximation algorithm and our ACO approach. The third section shows the cost of the solution returned by both algorithms. The final section shows the number of generations ACO took to converge to its solution.

Table 4. Covering Roadway Scenarios.

	US 299 (a)	US 299 (b)	US 3	US 199	US 191
Segment length	16.6km	22.2km	16.5km	12km	39.3km
Vertices	1329	1771	1317	956	3140
Edges	41,307	375,626	74,301	24,157	225,497
ITM algorithm	24.4s	48.1s	25.2s	13.4s	384.6s
Floyd's algorithm	48.8s	125.2s	48.0s	17.3s	647.0s
2-approximation	5.0s	20.7s	5.8s	2.8s	37.5s
ACO algorithm	214.1s	422.3s	225.3s	150.8s	763.6s
Approximation cost	$211,350	$140,770	$194,320	$128,960	$275,146
ACO cost	$140,350	$126,770	$152,330	$89,970	$236,152
ACO generations	11	13	7	8	2

Portions of the ACO solution for the US-199 (b) scenario are shown in Figure 6, and for the US-191 scenario in Figure 7.

For the five scenarios tested, ACO performed 22% better than the 2-approximation algorithm on average and 34% better at best when comparing solution costs. ACO resulted in a total savings of $204,974 when compared against 2-approximation costs.

ACO converged to its solution after 8 generations on average and 13 generations at worst. Running ACO for more than 16 generations might have found better solutions, but these results show that good solutions can be found quickly.

While both algorithms run in polynomial time, the actual running time of ACO was 35 times longer than the 2-approximation. Comparing the average benefit $40,994 against the average running time of about 6 minutes, ACO clearly offers an advantage over the 2-approximation algorithm.

Adding Bandwidth and Delay Constraints

We then modified the roadway-covering NDP described above to have bandwidth and maximum delay constraints. We tested the extreme case where each discretized point along the roadway would have a bandwidth requirement of 64kbps and a maximum delay requirement of 100ms.

We tested both the US-299 (a) and US-199 (b) scenarios, running ACO for 16 generations of population 64 each. The results are summarized in Table 5.

Table 5. Covering Roadway Scenarios with Constraints.

	US-299 (a)	US-199
Original ACO algorithm	214.1s	150.8s
Constrained ACO algorithm	393.5s	360.6s
Approximation cost	$211,350	$128,960
Original ACO cost	$140,350	$89,970
Constrained ACO cost	$221,380	$120,970
ACO generations	9	11

We cannot directly compare constrained ACO results against the 2-approximation algorithm because it does not consider bandwidth or delay constraints. We mention our earlier ACO and 2-approximation results for reference only.

For the scenarios tested, we see that a constrained ACO algorithm runs about 2 times slower than an unconstrained ACO algorithm. This is because constrained ACO algorithms are more likely to become trapped in dead-end cases where all edges with enough available bandwidth have been exhausted (and thus are tabu). As described in our methodology, solutions are reinitialized to $S = (\emptyset, \emptyset)$ when they become trapped.

Although we have no direct comparison, we can see that constrained ACO solutions are still near the costs of the unconstrained ACO and 2-approximation algorithms. The constrained ACO algorithm converges after 10 generations on average, compared to 8 generations average for the unconstrained ACO algorithm.

Figure 8 shows the MST solution selected by the unconstrained ACO algorithm for the US-199 (b) scenario, and Figure 9 shows the MST solution selected by the constrained ACO algorithm for the same scenario.

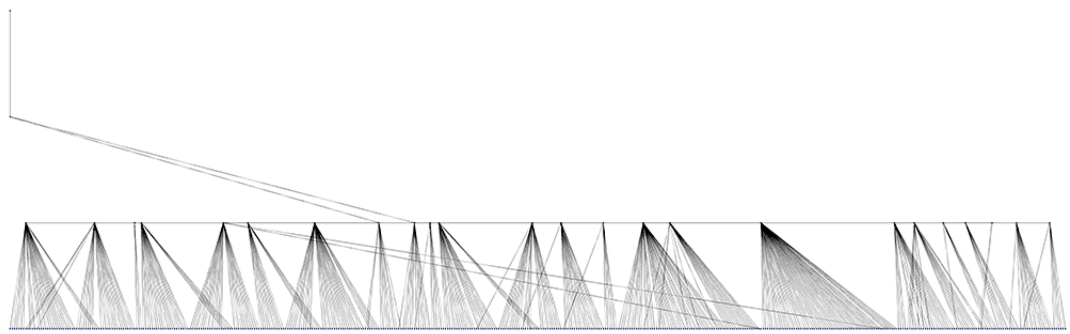


Figure 8. Unconstrained MST Solution.

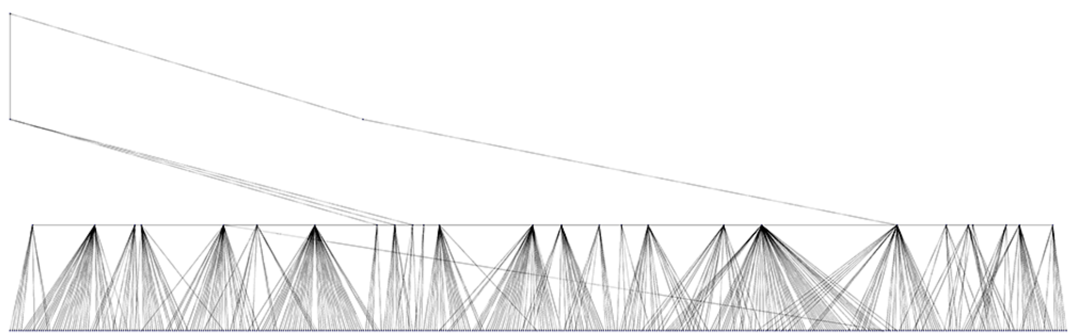


Figure 9. Constrained MST Solution.

We can see in Figure 9 that the constrained ACO algorithm is forced to include more routes as it works towards the global backbone at the top of the GSTS graph. In the constrained MST solution, bandwidth aggregates as ants make their way towards the global backbone. As the bandwidth is exhausted from previously selected links, ants select other links that allow them to still reach the global backbone.

The unconstrained MST solution only selects two radio links from roadside relays to a single backbone relay. In Figure 9 the constrained ACO algorithm selects one additional backbone relay and two additional links from the roadside to the backbone. In addition, much of the roadside relay structure is reorganized to distribute traffic loads without violating bandwidth constraints.

These additional links allow the solution to meet the bandwidth and delay constraints while still minimizing network cost. In both scenarios our ACO approach quickly converges to solutions that are similar in cost to those found by the unconstrained ACO algorithm.

β Analysis

One important variable of our ACO algorithm is β which weights our local heuristic. Optimal β values will help ACO find good solutions quickly, while non-optimal values will increase the time needed to find good solutions. Setting $\beta = 0$ is equivalent to a random search without heuristic guidance. In this section we present extensive testing of various β values before selecting $\beta = 30$.

For the roadway-covering NDP described earlier we tested four scenarios US-199 (a), US-199 (b), US-191, and US-101. For each scenario, we independently tested discrete β values in the range [1,120]. For each β tested, we ran our ACO algorithm 15 times and compared the average solution cost against the 2-approximation algorithm. While the 2-approximation solution is constant across all β, each ACO run is different due to the randomness involved.

Below we present each of the four scenarios, showing the average ACO solution cost with standard deviation. The left axis is the solution cost in thousands of dollars, and the right axis is the average number of generations that each ACO run took to converge to its solution.

In Figures 10, 11, 12, and 13 we see that higher β values consistently lead to better ACO solutions. Beyond $\beta = 20$ we see little improvement in solution quality.

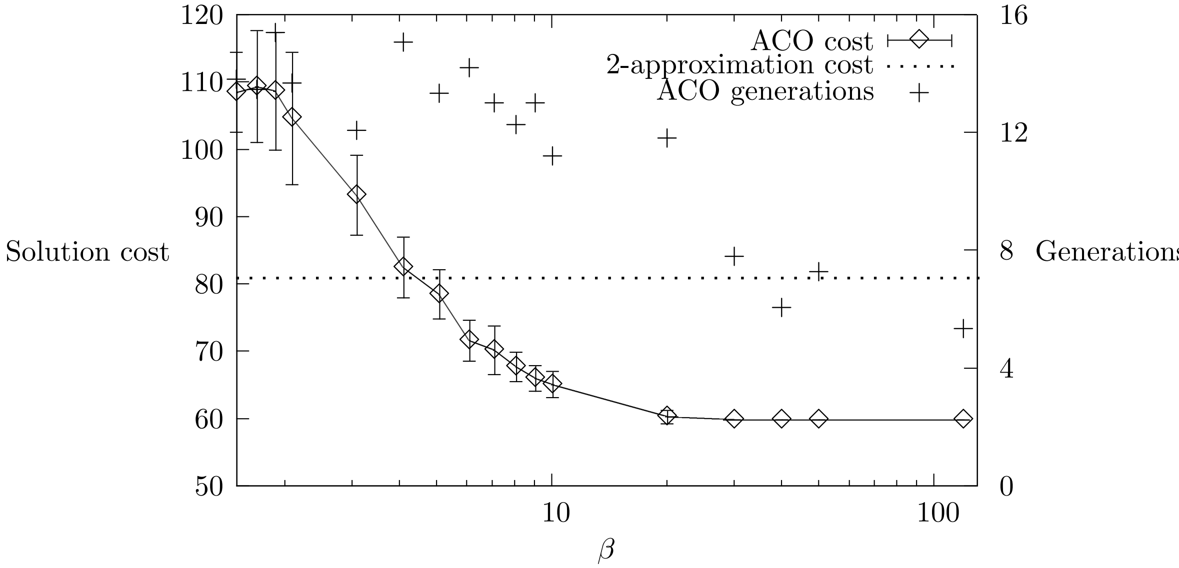


Figure 10. β Analysis for US-199 (a) Scenario.

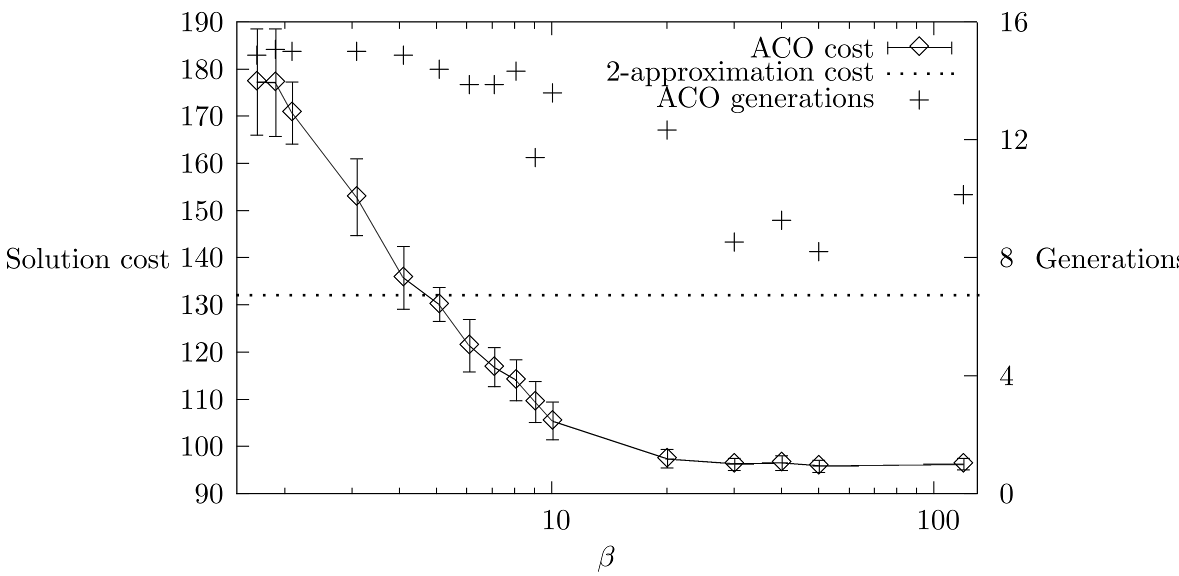


Figure 11. β Analysis for US-199 (b) Scenario.

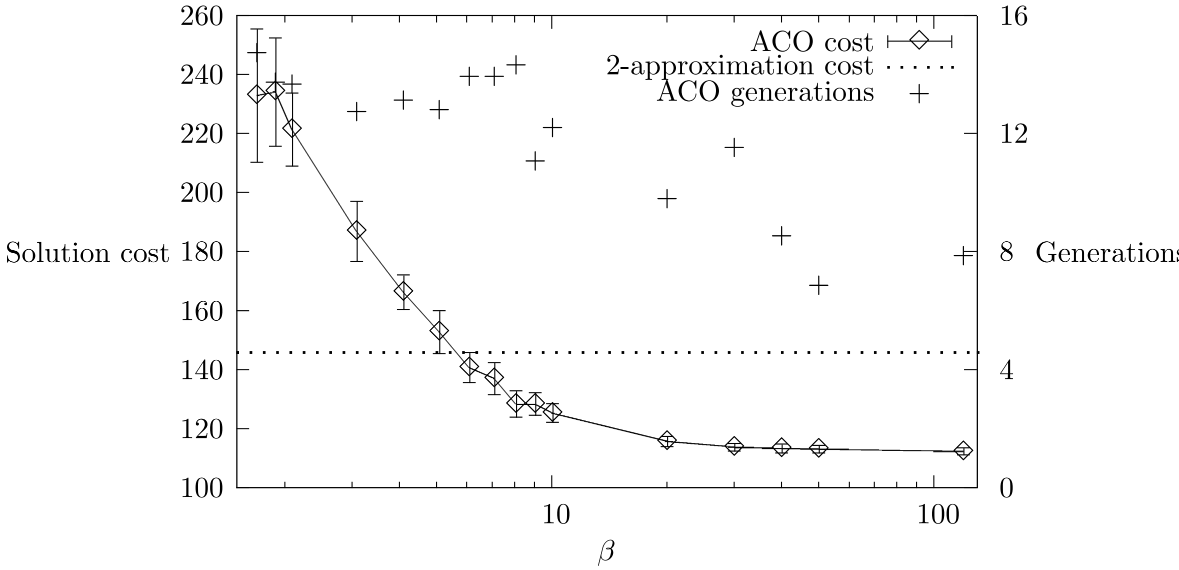


Figure 12. β Analysis for US-191 Scenario.

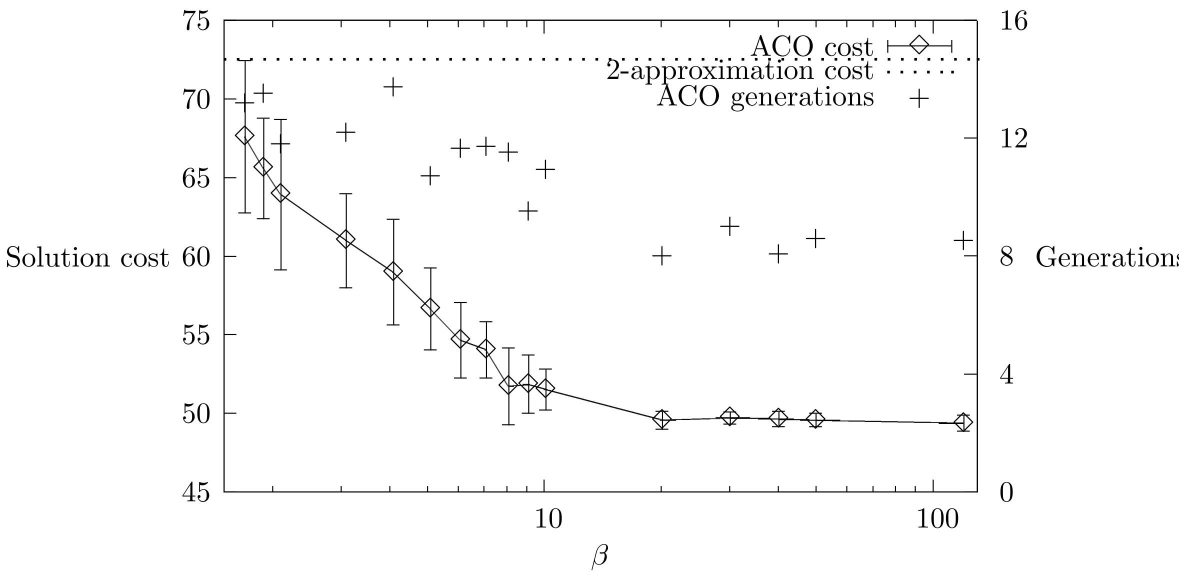


Figure 13. β Analysis for US-101 Scenario.

When comparing average generations to convergence in solution, we see that higher β values tend to converge faster. However, in Figures 11 and 12 we see that $\beta = 120$ takes more generations on average to converge than lower β values.

We conclude that β values beyond 20 offer little improvement to solution quality, and that $\beta \geq 120$ may increase convergence time. We see in each scenario that $\beta = 30$ offers a good trade-off between solution quality and convergence time. Therefore, we use $\beta = 30$ for our ACO algorithm.

CONCLUSIONS

In this thesis we developed a generalized Steiner tree-star (GSTS) construction for solving the network design problem (NDP). We then developed an ant colony optimization (ACO) approach to solving the minimum Steiner tree (MST) problem on this GSTS graph while meeting a wide range of constraints including coverage, bandwidth, delay, and redundancy. This is the first approach to date that considers all of these constrains simultaneously while minimizing network cost.

We tested our GSTS construction and ACO algorithm using several real-world scenarios. Our approach quickly found almost-optimal solutions for the two relay-along-roadway NDP scenarios when compared directly with known optimal solutions.

For the five roadway-covering NDP scenarios, our ACO approach performed, on average, 22% better than the 2-approximation algorithm. Our approach had a $40,994 benefit over 2-approximation solutions on average with a running time of under 15 minutes in each case. Our approach converged to its solution after 8 generations on average. Running ACO for more than 16 generations might find better solutions, but our results show that relatively good solutions are found quickly.

When applying bandwidth and maximum delay constraints to two roadway-covering NDP scenarios, our ACO approach finds solutions similar in cost to the unconstrained NDP. No direct comparison can be made to the 2-approximation algorithm because it does not consider bandwidth or maximum delay constraints.

We investigated the impact of β on our metaheuristic formulation, and found that $\beta = 30$ offers a good trade-off between solution quality and convergence time. β values above 30 offer only minimal quality improvement with a possible increase in convergence time.

This thesis shows that our GSTS construction and ACO approach can find near-optimal NDP solutions in polynomial time. In addition, our approach is the first to simultaneously consider coverage, bandwidth, maximum delay, and redundancy constraints.

Future Work

Future work should compare ACO solutions against networks designed by human engineers. This will give another measure of the benefit offered by our ACO approach in addition to the 2-approximation algorithm presented here.

Radio engineers often work to maximize the margin between receiver sensitivity and signal power, represented in Equation 3.2. Larger margins make radio links more robust when faced with interference not considered by the ITM propagation algorithm. Our ACO approach could reach this goal by modifying the local heuristic to prefer links with higher margins.

The network routing problem (NRP) searches for optimal routes through an existing network in light of available links and bandwidth requirements. These routes are then used to deliver traffic across the network. Because our approach considers static bandwidth requirements, the routes chosen by our ants could be used as initial static routes in a dynamic NRP algorithm. Other work has applied ACO to solve the dynamic NRP [38] [39].

REFERENCES CITED

[1] F. Ikegami, S. Yoshida, T. Takeuchi, and M. Umehira. Propagation factors controlling mean field strength on urban streets. *IEEE Transactions on Antennas and Propagation*, 32:822–829, 1984.

[2] R. Kahn. The organization of computer resources into a packet radio network. *IEEE Transactions on Communications*, 25(1):169–178, 1977.

[3] A. Markopoulou, F. Tobagi, and M. Karam. Assessment of VoIP quality over internet backbones. pages 150–159, 2002.

[4] P. Calégari, F. Guidec, P. Kuonen, B. Chamaret, S. Udeba, S. Josselin, and D. Wagner. Radio network planning with combinatorial optimisation algorithms. *ACTS Mobile Communications*, pages 707–713, 1996.

[5] Samir Khuller and An Zhu. The general steiner tree-star problem. *Information Processing Letters*, 84:215–220, 2002.

[6] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. B. Shmoys, É. Tardos, and D. P. Williamson. Improved approximation algorithms for network design problems. In *SODA '94: Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 223–232, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.

[7] M. Goemans and D. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Mathematical Programming*, 60:145–166, 1993.

[8] Bin Hao, Han Tang, and Guoliang Xue. Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation. *High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on*, pages 246–250, 2004.

[9] Jian Tang, Bin Hao, and Arunabha Sen. Relay node placement in large scale wireless sensor networks. *Computer Communications*, 29(4):490–501, 2006.

[10] Sudeept Bhatnagar, Samrat Ganguly, and Rauf Izmailov. Design of IEEE 802.16-based multi-hop wireless backhaul networks. In *AcessNets '06: Proceedings of the 1st international conference on Access networks*, page 5, New York, NY, USA, 2006. ACM.

[11] Mamadi Diané and Ján Plesník. An integer programming formulation of the steiner problem in graphs. *Mathematical Methods of Operations Research*, 37(1):107–111, 1993.

[12] Xiuzhen Cheng, Ding-Zhu Du, Lusheng Wang, and Baogang Xu. Relay sensor placement in wireless sensor networks. *Wireless Networks*, 2007.

[13] Ajit Agrawal, Philip Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized steiner problem on networks. pages 134–144, 1991.

[14] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica*, 15:141–145, 1981.

[15] H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Math. Jap.*, 24:573–577, 1980.

[16] Stefan Hougardy and Hans Jürgen Prömel. A 1.598 approximation algorithm for the steiner problem in graphs. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 448–453, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.

[17] Marek Karpinski and Alexander Zelikovsky. New approximation algorithms for the steiner tree problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 2(30), 1995.

[18] Gabriel Robins and Alexander Zelikovsky. Improved steiner tree approximation in graphs. In *SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 770–779, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[19] P. Calegari, F. Guidec, P. Kuonen, and D. Wagner. Genetic approach to radio network optimization for mobile systems. *Vehicular Technology Conference, 1997 IEEE 47th*, 2:755–759, 1997.

[20] H. Meunier, E.-G. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimization. *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 1:317–324, 2000.

[21] M. Shahbaz. Fixed network design of cellular mobile communication networks using genetic algorithms. *Universal Personal Communications. 1995. Record., 1995 Fourth IEEE International Conference on*, pages 163–167, 1995.

[22] R.M. Whitaker, L. Raisanen, and S. Hurley. A model for conflict resolution between coverage and cost in cellular wireless networks. *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, 2004.

[23] D.B. Jourdan and O.L. de Weck. Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, 5:2466–2470, 2004.

[24] S. Soni, S. Narasimhan, and L.J. LeBlanc. Telecommunication access network design with reliability constraints. *Reliability, IEEE Transactions on*, 53(4):532–541, 2004.

[25] S.M. Allen, S. Hurley, and R.M. Whitaker. Automated decision technology for network design in cellular communication systems. *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, 2002.

[26] Benoit Lardeux, Arnaud Knippel, and Jerome Geffard. Efficient algorithms for solving the 2-layered network design problem. *Proceedings of the International Network Optimization Conference*, pages 367–372, 2003.

[27] Jiefeng Xu, Steve Chiu, and Fred Glover. Using tabu search to solve the steiner tree-star problem in telecommunications network design. *Telecommunication Systems*, 6(1):117–125, 1996.

[28] Nejla Ghaboosi and Abolfazl T. Haghighat. A tabu search based algorithm for multicast routing with QoS constraints. *9th International Conference on Information Technology (ICIT'06)*, 0:33–39, 2006.

[29] S. Gosavi, S. Das, S. Vaze, G. Singh, and E. Buehler. Obtaining subtrees from graphs: an ant colony approach. *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pages 160–166, 2003.

[30] Yu Hu, Tong Jing, Xianlong Hong, Zhe Feng, Xiaodong Hu, and Guiyang Yan. An efficient rectilinear steiner minimum tree algorithm based on ant colony optimization. *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on*, 2:1276–1280 Vol.2, 2004.

[31] Yufeng Xin, Tuna Guven, and Mark Shayman. Topology design for wireless adhoc networks with backbone support. *Conference on Information Sciences and Systems*, 2005.

[32] Nicholas De Minco and Christopher Holloway. Propagation modeling for its applications in the roadway environment. *Journal of Intelligent Transportation Systems*, 3(4):287–331, 1997.

[33] Scott Rose. The effect of digital elevation model resolution on wave propagation predictions at 24GHz. Master's thesis, Virginia Polytechnic Institute and State University, 2001.

[34] M. Al-Nuaimi and A. Hammoudeh. Measurements and predictions of attenuation and scatter of microwave signals by trees. *Microwaves, Antennas and Propagation, IEE Proceedings*, 141(2):70–76, 1994.

[35] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. Bradford Book, 2004.

[36] Thomas Stützle and Holger Hoos. Improvements on the ant system: Introducing max-min ant system. In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 245–249, 1997.

[37] Thomas Stützle and Holger Hoos. The max-min ant system and local search for combinatorial optimization problems. In *Proceedings of the 2nd International Conference on Metaheuristics*, pages 313–329, 1999.

[38] M. Gunes, U. Sorges, and I. Bouazizi. Ara-the ant-colony based routing algorithm for MANETs. In *Proceedings of the International Conference on Parallel Processing*, pages 79–85, 2002.

[39] Gianni Di Caro and Marco Dorigo. Antnet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:317–365, 1998.