Summary

Our task is to construct an efficient network of radio repeaters that can accommodate 1,000 simultaneous users on a spectrum from 145 to 148 MHz in a circular, flat area of radius 40 miles. In the United States, this frequency range is used by licensed amateur "ham" radio operators. Because the United States government requires ham radio operators to undergo a certification process, we assume that we can implement protocols that the users will follow when using the network. Our goal is to minimize the number of repeaters used by the network while also providing a reasonable amount of service to users.

We approach this problem by trying to construct a network that places more repeaters in regions with the highest population density. We utilize privacy lines in a systematic way to allow parallel communication and minimize interference across our network. To form the basis of our network, we use a clustering algorithm to group points in areas of dense population and then place repeaters at the centroids of each region. If the resulting graph is disconnected, this preliminary network is connected using a minimum spanning tree algorithm.

In order to test our network we came up with sample population data using a clumped distribution model. The model produces data that had a similar distribution to actual density data taken from a flat region in the Southeastern United States. We measure the coverage of our network by the percentage of users within range of our network's repeaters. We calculate the transmission capacity of the network is through repeated simulations of users in the network attempting to send messages to other users in the network. Sensitivity analysis was performed on this to determine the accuracy of this metric to be within 1.5% error. Our model was able to significantly increase the transmission capacities up to 98% transmission over the network in our model. This compares to a maximum of 32% for a naive approach of evenly distributing repeaters throughout the area.

We provide color-gradient plots for different population distributions and plots of our generated networks for these distributions in our paper. Our algorithm works very well for most clumped populations. However, it is limited in the extent to which we can cluster the data. In addition to this information, we provide a method for constructing subnetworks around nodes with high load, a strategy that can improve the ability of the network to accommodate larger numbers of users.

Clustering on a Network

MCM Contest Question B

Team # 10496

February 14, 2011

Contents

1 Introduction

1.1 Background

The goal of this paper is to present an extendible, flexible algorithm for the optimal placement of a repeater network intended to serve users along the 145-148MHz range. In the United States, this particular band of frequencies corresponds to those used by amateur "ham" radio users [6]. Because of how radio repeaters work, interference is a significant, persistent problem. With more than 700,000 licensed amateur radio operators in the United States, there is a real need to further develop strategies to optimize the number of repeaters needed to service an area as well as methods for managing interference [3]. This need is especially acute in extremely flat areas, where the repeaters' limited maximum range requires chains of repeaters to be used.

Technological Background

Radio waves in the 145-148MHz range are considered very high frequency (VHF). Very high frequency waves are often used for short-range radio communication. For one thing, they are generally not reflected much by the ionosphere, so communication via VHF waves is largely limited to slightly more than line-of-sight communication. The line-of-sight property is so important that in a flat area, you can make a reasonable approximation of a transmitter's broadcast radius simply based on its height. A rough approximation of the line-of-sight horizon (in miles) for an antenna of height h (in feet) is:

$$distance = \sqrt{1.5 \cdot h}$$

[4]

While the limited range of VHF transmissions provides one avenue for avoiding interference with other transmissions, it also limits the broadcasting range of an individual user. On flat ground, the broadcast radius of an average handheld amateur radio is somewhere in the neighborhood of 3-5 miles [2]. For amateur radio enthusiasts who want to be able to communicate with other people like them, this is an extremely limited range.

To address this issue, amateur radio users can tap into any of hundreds of radio "repeaters" distributed across the country. What a repeater does is clear from its name: it repeats the signals it receives, often with additional power or from a higher vantage point. The type of repeater available for our use is called a "duplex" repeater. The repeater is tuned to pick up a particular frequency (in the 145-148MHz range), and then outputs the same signal at a frequency offset by 600KHz. Because repeaters have a much larger range than the average user due to their increased height, repeaters allow an amateur radio operator broadcast to a much wider audience.

The ability to broadcast across a much wider area is a positive one for amateur radio operators, but it does present certain challenges. Most importantly, offering users a greater broadcast range greatly amplifies the likelihood of interference. One tool that has been used to address the problem is the ”continuous tone-coded squelch system” (CTCSS) technology. The idea is clever and simple: each repeater is associated with a particular subaudible tone. A given repeater remains in a deactivated state until it hears a signal that is of the proper frequency *and* contains its subaudible tone. As long as the repeater hears the signal with the subaudible tone, it will remain active and will rebroadcast any signals it receives at its input frequency.

1.2 The Problem

We have been tasked with determining the minimum number of repeaters needed to accommodate a particular number of users - initially, 1000 - spread across a flat, circular area of radius 40 miles. In part because we are dealing with the relatively narrow range of 145-148 MHz, we have 54 distinct privacy lines at our disposal to help mitigate potential interference problems.

The problem statement mentions that this network must be able to accommodate a certain number of “simultaneous users”. This is a vague phrase that is open to a number of interpretations. One way to interpret it is to say that there are 1,000 people making demands on the radio network at any given time. For a network of repeaters used for amateur radio, this is completely unrealistic from an interference standpoint. As a consequence, for the purpose of this problem, we defined ”1,000 simultaneous users” to mean “1,000 people who might make demands on the system”, where the average number of requests per user is a parameter of our model.

2 Assumptions

Over the course of our analysis, we make a number of assumptions both to simplify the problem and to more clearly define it. We believe that the majority of these assumptions are reasonable and based on solid real-world justification; we will note when we deviate from this practice.

First, over the course of this paper we are assume that both repeaters and individual users broadcast in a radially symmetric manner. That is, their broadcast range can be represented by a circle on a two-dimensional map. Given the fact that transmissions in the VHF range are largely dependent on having a line-of-sight connection, this makes sense for a flat area - transmission range is going to be limited mostly by the curvature of the earth, something that is roughly constant in each direction. In addition, we will make the assumption that if the broadcast radii of two repeaters touch, then they are able to communicate with one another. This is a reasonable assumption as well — the broadcast radius measures how far a source can broadcast to a point on the ground. Provided that power is not an issue and the radio waves don't fade too much with distance, the distance a 40-foot-high antenna will be able to broadcast to another 40-foot-high tower will be roughly twice its broadcast range [9].

Second, we assume that the radio waves are traveling instantaneously. Because radio waves travel at near the speed of light, this is not a huge distortion, especially at scales of eighty miles or less [8]. While a wave that needs to travel through several repeaters to get to its destination will suffer a noticeable delay, it will not be more than a second or two. In practice, this requires users to build in a buffer between receiving and transmitting signals. For the purpose of analyzing interference between two 30-second or minute-long signals, this buffer is rather small. As a consequence, we will use a set interval and ignore any requisite buffer when running any network load simulations.

Third, any repeater we use can have a maximum range of about 9 miles. This corresponds to an antenna height of about 40 feet [9]. Across a flat area that does not have any significant rises upon which to place an antenna, this is a reasonable maximum antenna height. An "average" user — presumably with in possession of a radio at least as powerful as a handheld amateur radio — is assumed to have a radial broadcast distance of roughly 3 miles. For the purposes of this paper, we assumed that a handheld radio could transmit a maximum of 5 miles to another handheld and assumed that a handheld radio could only transmit to a repeater if that repeater could reach the handheld radio. To account for power constraints on these mobile devices that could cause degradation of waves over large distances [4], we rounded down to 3 miles when considering the maximum transmission distance from a handheld device.

Fourth, the very real threat of interference as well as amateur radio licensing requirements have led to very widely-adopted broadcasting codes of conduct [10]. Before broadcasting on a particular frequency or using a nearby repeater, users are required to listen on all privacy lines for several minutes to see if it is in use. If it is not, then the user will make his or her broadcast. Therefore, during analysis, we assumed that users are both capable and polite. That is, they do not make requests when a local tower is in use.

Fifth, while this rules out the possibility of interference at a local level, it does not rule out the possibility of interference across a long chain of connected repeaters. Ideally, this type of interference would count as a ”failed attempt to communicate”. However, to simplify and accelerate the model-development process, we made a temporary assumption that, due to time constraints, ended up becoming permanent. Instead of counting the interfering signals as ”failed” requests, we simply make the network unable to process two conflicting requests at the same time. So while this type of interference counts against a network, it isn’t penalized as much as it should be.

Sixth, it makes sense to think about a minimum spacing such that two frequencies are clearly distinguishable from one another. In our case, we simply went with the information that was given to us. Presumably, there is a good reason that the repeaters we are given output a signal that is offset 600KHz from the signal they receive. For the purposes of this paper, we assume that it is because 600KHz is a good minimum discrepancy.

Seventh, we assume that each user places a roughly equal strain on the network. While this would certainly not hold true in the real-world, over a large body of users things will average out and this will tend to be a reasonable approximation.

Eight, we assume that the location we are dealing with is in the United States. This assumption is what tells us that we are in fact dealing with constructing a repeater network for use by amateur radio operators.

Finally, we make the assumption that the distribution of calls being made is strongly biased toward local connections. However, we feel that it is also a very reasonable one. In the real world, extended chains of repeaters have very difficult issues managing interference — unless the signal is passing solely through a user’s local repeater, there’s no way to tell when it will be possible to make a call to a distant location. As a consequence, linked chains of repeaters are most frequently used for public service announcements during emergencies or in areas where there is a very limited amount of demand CITE. Simply looking at the numbers, it is extremely unlikely that it is possible to accommodate a large number of users unless each makes a minimal demand on the network as a whole. In this light, a heavy bias toward calls that use a small number of repeaters is reasonable.

3 Metric

3.1 Measuring Transmission Capacity

In order to get some sense for whether or not a given solution was valid, we needed to develop some way to measure a given solution's efficacy. More importantly, for a given network we needed to be able to return a yes or no answer for whether or not that network "accommodates" its users. In order to develop a reasonable metric, we first asked "what, fundamentally, does an amateur radio operator want a repeater network to do?" At a basic level, the answer is very simple: an amateur radio user is going to want to broadcast something. The user will be satisfied if he or she is able to do this in a reasonable, timely manner, with a minimum number of interference-related issues.

With this definition of accommodation in mind, there is a very straightforward way to define our metric. One network is better at accommodating its users than another if it can successfully process more transmission attempts than another. Similarly, we can designate a network as satisfactory if it can handle a minimum number of transmission attempts over a given time period. Expressing this as a ratio, we have:

$$Q = \frac{\#\ \ transmissions\ \ processed}{\#\ \ transmissions\ \ attempted}$$

Note that requests made by or to a user that is not covered by the network are automatically counted as "failed" attempts at communication. The pseudocode in Algorithm 1 describes how we calculated Q.

Algorithm 1 Measure Network Quality

Require: $UserPositions$, $UserRepeaterAssignment$, $UserTransmissionNumber$, $SimulationLength$, $NetworkAdjacencyMatrix$, $UserRanges$

Ensure: $UnfulfilledTransmissions$ equals the number of desired user transmissions that could not be completed during the simulation.

Run $GenerateUserTransmissions()$ with inputs $UserRepeaterAssignments$, $UserTransmissionNumber$ and $SimulationLength$ to get $Transmissions$, which represents where each user will transmit to and when they will try to do so.

Set $TransmissionQueue$ be an empty array.

Set $UnfulfilledTransmissions$ to zero.

for all timesteps in $SimulationLength$ **do**

  for all Transmissions in $Transmissions$ **do**

    If this request is scheduled for this timestep, add it to $TransmissionQueue$

  end for

  Set $CurrentTransmissions$ to be an empty array.

  for all Transmissions in $TransmissionQueue$ in order **do**

    if this transmission is being made by or to a user who is not covered by the network **then**

      Increment $UnfulfilledTransmissions$

      Remove this transmissions from $TransmissionQueue$

    else if this transmission will not interfere with any transmissions in $CurrentTransmissions$ **then**

      Add this transmission to $CurrentTransmissions$ and remove it from $TransmissionQueue$.

    end if

  end for

end for

Add the number of elements remaining in $TransmissionQueue$ to $UnfulfilledTransmissions$.

Though the metric itself is pretty straightforward, it is less clear what exactly a reasonable proportion of successful communications should be required in order to consider a repeater network successful. As a result, our model has a parameter for Q. This allows someone who has a better idea for how accommodating they need the network to be to set it at whatever threshold they desire.

3.2 Generating Requests

In order to generate requests for using the network we need to assign a starting location and an ending location for this request. Typically people are more likely to make requests closer to their current location since they will have more interaction with people locally then over a long distance. To model this type of distribution of requests by a given person, we use a Poisson distribution to calculate the probability of a call going a distance l between two towers,

$$\binom{n}{l}\frac{\lambda^k}{l!}\left(1-\frac{\lambda}{n}\right)^{n-l}.$$

Here, n corresponds to the maximum number of towers that one can send a request. The number of towers was used for calculating this since people with closer towers will generally have shorter Euclidean distances given us the desired effect. In addition to this, amateur radio users might consider the amount of network they are using for a call which would also result in a decay as l increases. By choosing $\lambda = 1$ to provide the desired decay of the requests as l increases.

3.3 Measuring Overall Quality

There is more that goes into determining the overall quality of a network than whether or not it is able to handle a certain percentage of user requests. For example, there are a huge number of distinct networks (infinite if we don't specify a grid of coordinates) that will all satisfy the minimum accommodation requirements. We need to have some method to distinguish between two "passing" networks. Fortunately, the problem statement makes it very clear what the primary criterion should be, saying: "determine the minimum number of repeaters necessary to accommodate 1,000 simultaneous users". To address this, we examined the solutions generated by our algorithm for a particular number of maximum repeaters used. Using this approach, it is easy to find a lower bound for the number of repeaters needed to adequately accommodate present users with a network generated by our algorithm. Other factors that might be worth considering when comparing the overall quality of networks include factors such as coverage percentage and how they perform with different levels of demand.

4 Approach

4.1 Modeling Population Density

Our preliminary model for modeling the population distribution within our region was to simply randomly locate 1000 people inside of our region with a uniform distribution. However, humans are social animals, so assigning each person a random location independently of each other is a poor model for what actual population distributions look like. It is more common to model a human population by using a clumped population distribution. This involves people aggregating around certain areas which is common in the formation of communities. Population data on the density of people in different communities suggests that within different regions we will have different population size [5].

We modeled different types of population distributions that can occur by choosing a set number of clumps that occur within our region. We used a variety of different numbers of clumps to account for different distributions that can occur depending on the given region. The mean of a each clump was selected by generating a random point inside the circle. For a given clump we modeled the distribution of the population around the center by using a multivariate Gaussian distribution. Since the x and y coordinate of the person's location are independent of each other, we used the multivariate distribution:

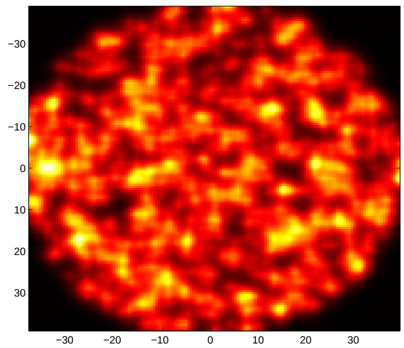
$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\left[\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right]}.$$

We used values for $\sigma_x = \sigma_y = 20$, which seemed to provide a reasonable range of models for the data. This was determined by looking at actual data for population density in flat areas of the United States and comparing our density distributions to the actual distribution [1].

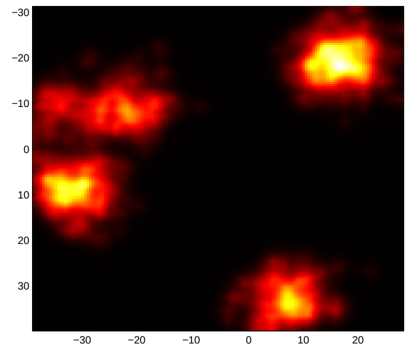
4.2 Linking Strategies

4.2.1 Two Different Strategies

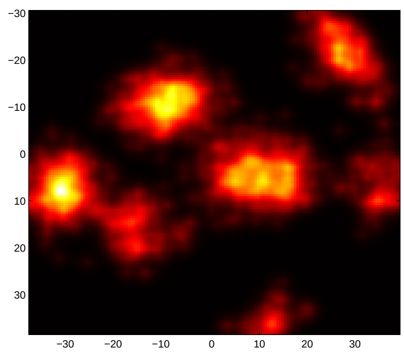
Before jumping into our overall approach, it makes sense to talk a little bit about the methods at our disposal for creating a chain of repeaters allowing for two-way communication. In order for a repeater to be communicate to another, its transmission frequency must be the same as the



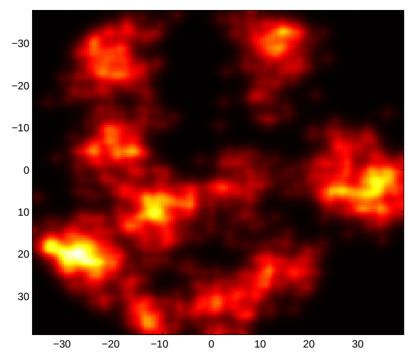
(a) Random Population Distribution



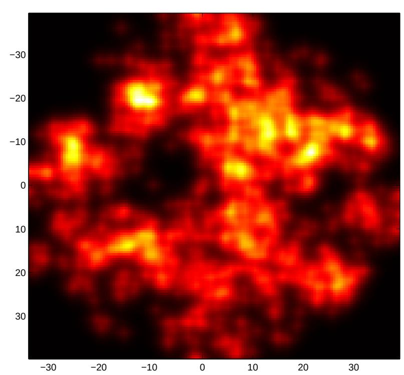
(b) Clumped Population Distribution with 5 Clumps



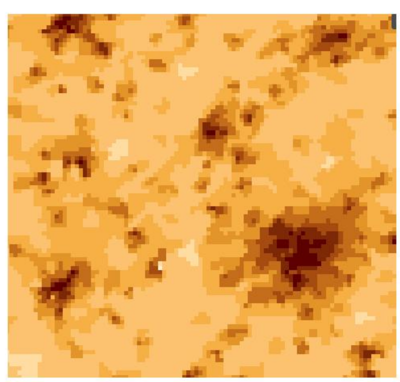
(c) Clumped Population Distribution with 10 Clumps



(d) Clumped Population Distribution with 20 Clumps



(e) Clumped Population Distribution with 40 Clumps



(f) Actual Population Distribution

Figure 1: Pictures of the different population distributions we generated and a comparison to the actual population distribution taken from the Southeastern United States [1]

other's receiver frequency. The most basic way to create two-way capability is to, for example, give receiver A a receiving frequency of 145MHz and a transmission frequency of 145.6MHz and have receiver A′ be the inverse of that (145.6MHz receiving, 145.0MHz transmission). A deeper analysis shows an immediate, serious flaw with this approach. When a signal is being transmitted from one to the other, it will be (essentially) immediately transmitted back to the first, creating an feedback loop of transmission. As a repeater is actively receiving whenever it is transmitting, there will be some amount of feedback even if we attempt to implement privacy lines in some clever way.

The other approach to creating a chain of is to ”stagger” the receiving and transmission frequencies such that the first repeater in a chain transmits at the frequency the second chain receives on, and so on. From our assumed minimum spacing of 600KHz, we have 5 distinct pairs of frequencies to work with. Let us assign names to the types of repeaters as follows:

- A - receiving frequency = 145.0MHz, transmitting frequency = 145.6MHz
- B - receiving frequency = 145.6MHz, transmitting frequency = 146.2MHz
- C - receiving frequency = 146.2MHz, transmitting frequency = 146.8MHz
- D - receiving frequency = 146.8MHz, transmitting frequency = 147.4MHz
- E - receiving frequency = 147.4MHz, transmitting frequency = 148.0MHz

Using these sets of frequencies, we see that a chain like A-B-C-D-E is a viable method for providing more long-range communication. Indeed, this method of chaining is the one we adopt in our approach to the problem.

4.2.2 Bi-directional and Extended Chains

One issue yet to be addressed by the chaining method described above is how one can transmit in the opposite direction. An effective way of dealing with this issue - and the one we've adopted for the rest of this report - is using the ”inverses” of the repeaters enumerated above to transmit in the opposite direction. If we pair an A′ repeater with every B, B′ with C, C′ with D and D′ with E, then we can get two-way transmission along this line. Without using private lines in a clever way, we still run into the feedback issue we ran into above - we'll describe how we plan to get around this below.

In addition to bi-directionality, it's not immediately clear from the above description how to create a chain of repeaters longer than 5. First, given the size of area involved, the range of the repeaters being used, and how each repeater is assigned a type, it's unlikely there will be very many longer chains. However, in the interest of covering extreme cases and allowing this algorithm to be generalized to larger regions, we provide a method for creating longer chains. We could create a very weak-broadcasting E′ tower at the edge of the range of the end of an A-B-C-D-E chain (weak enough that its transmission back to the E repeater is not an issue), pairing it with both a D and a D′ repeater at the same location. The D and D′ repeaters would be set to different privacy lines so they wouldn't be activated simultaneously - if they are activated simultaneously, we can simply shut them both down or have the repeaters time out after a fixed amount of time. After this point, the chain could continue to be extended through linking in a C/C′ node, then a B/B′ node, and so on.

4.3 Naive Approach

An extremely simplified way to view this problem is as follows: let us provide a way for anybody to broadcast anywhere by simply blanketing the entire area under consideration with a network of evenly-spaced, connected repeaters (and then removing those repeaters that don't cover any people). There are a number of issues with this approach. First, it's far from efficient - depending on the population distribution, it's very likely that at least some of the given area is at most sparsely populated, and unnecessary to cover with repeaters. Second, if it is implemented by alternating ″inverse″ repeaters (repeaters who can speak to each other), it's likely there would be significant issues in practice. Without using some kind of privacy lines, there would be no way to prevent some kind of feedback loop from occurring. Using a staggered repeater approach without running into the feedback issue would allow for only one-way transmission, something that would be unacceptable for most amateur radio use.

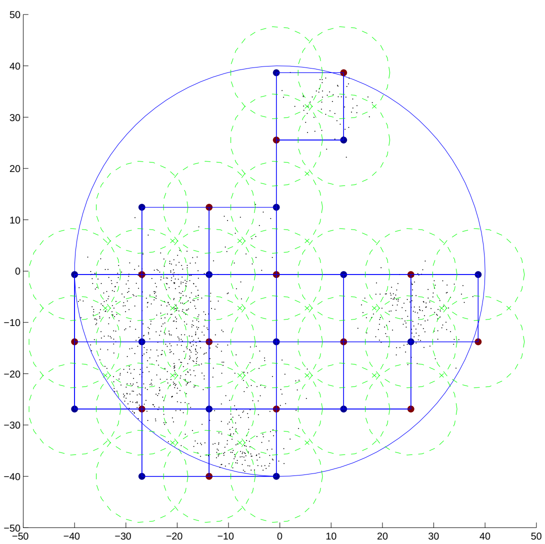


Figure 2: The network generated by the naive approach described above

Finally, even if there were no issue with feedback, a single, blanketing network would allow for very little usage. The user would have two options: either he or she could talk locally without using a repeater or he or she could broadcast to the entire area. As a consequence, only a single user could use the network at any given time. For a large number of users, this would be untenable.

4.4 Implementing Privacy Lines to Reduce Interference

As mentioned earlier in the document, using privacy lines can help mitigate interference. However, they need to be used carefully in order to be of much use. Two repeaters with differing privacy lines that are within range of one another can still interfere with one another if both are activated and broadcasting at the same time. Indeed, with the node layout above, there is really no “direction” of talk, and broadcasts take multiple paths to get to the same location. Because of these complexities, there’s really no way to assign privacy lines to organize communications in a controlled, rational manner.

One common method of organization for repeater networks is to assign one repeater to be the central“”hub” of the entire network CITE. Assigning a hub affords us a number of benefits. For one thing, it lets us assign a direction to each broadcast - each transmission can either be “toward” or “away from” the hub. In addition, it lets us create subnetworks across which users can talk without monopolizing the whole network, something that has a very important benefits. Along with our assumption about the distribution of calls, these subnetworks let us localize demand - a call between two people along a certain subnetwork can be entirely confined to that subnetwork, leaving the rest of the network free for others to use.

Being able to categorize a signal as moving in a particular “direction” does no good if we cannot actually control how the signal propagates through the network. Below is an image illustrating the method we use to achieve directional control:

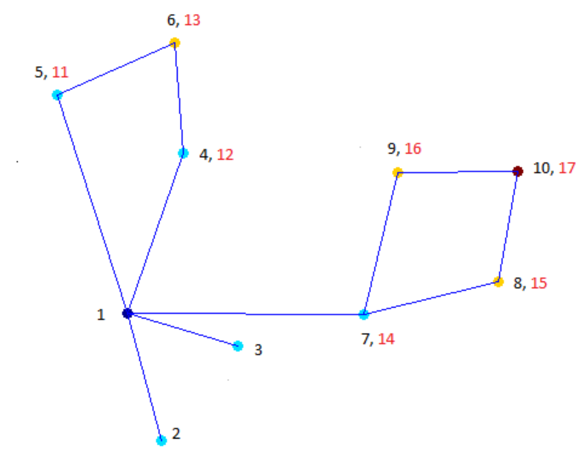


Figure 3: A typical network generated by our algorithm. The colors correspond to the particular type of repeaters present at that location. Dark blue = A, light blue = B/A′, yellow = C/B′ and maroon = D/C′.

As can be seen above, most repeaters have two privacy lines associated with them. This is a destination label rather than the particular privacy lines accepted there. The above system works as follows. First, the red numbers represent the privacy lines associated with incoming calls destined for other sub-branches of the network. For example, if repeater 6/13 wants to send a message to repeater 8/15, it would broadcast a signal on privacy line 15. Depending on which B/A' repeater it is primarily linked with one it is linked with, the signal from 6/13 will be picked up by both 5/11 and 4/12 and then rebroadcast with privacy line 15 (under our assumption of instantaneous transmission, this is OK. However, in practical applications, this is a potential issue, and is addressed in our ”Limitations” section). When it reaches the hub, the hub hears the signal and rebroadcasts it with the outgoing privacy line associated with repeater 15 (in this case, 8). The node 7/14 is the only B/A' node to recognize that privacy line, and so it is the only one to pick it up and rebroadcast it (again, with privacy line 8). This is heard by the 8/15 node, which in turn broadcasts a signal with privacy line 54, indicating that the transmission is not to be passed any further.

That example was a little bit long, so let us review the core idea. The location of each pair of repeaters is associated with two privacy lines, one ingoing and one outgoing. Outgoing repeaters (A-B-C-D-E) accept the privacy line associated with their current location as well as the outgoing privacy lines associated with locations that they are “on the way” to, from the perspective of the hub. Ingoing repeaters also accept the ingoing privacy lines associated with repeaters that they are “on the way” to. This includes locations on the way to the hub on their subnetwork as well as those repeaters on other subnetworks. Effectively, users are free to communicate within their subnetworks without having to worry about affecting broadcasts in other subnetworks. The net result is a reduction of interference and an increase in the network's overall capacity.

4.5 K-Means Clustering

One of the weaknesses of the naive algorithm is its failure to adapt to the population distribution that it is presented with. Heuristically speaking, the best way to cover as many users as possible with as few repeaters as possible is to center repeaters in areas with high population density. This approach is especially effective in realistic population distributions since users are likely to be clumped together in regions of high density.

For this reason, clustering is an important part of our repeater network generation algorithm (which is outlined in the Main Network Generation Algorithm, included below). We use the k-means clustering algorithm, which seeks to group n data points into k clusters (where k must be specified) so that each data point is assigned to the cluster with mean closest

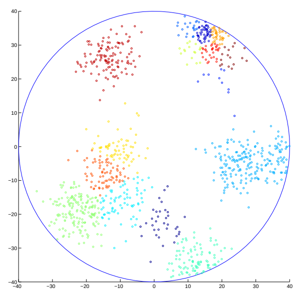


Figure 4: k-means clustering on a population

to its value, to assign each user to a cluster. Conceptually,*k*-means clustering assigns data points of similar value to the same cluster [7]. In our case, this means that users situated close together are grouped together. We then place a repeater in the mean location (or centroid) of each cluster. In general, this gives us a list of repeater locations located in regions of high density. In realistic population distributions, these repeater locations are likely to encompass population centers.

Algorithm 2 Main Network Generation Algorithm

Require: *UserPositions*, *MaximumRepeaterRange*, *UserRanges*, *NumberOfClusters*

Ensure: *RepeaterPositions* contains a list of all of the repeater positions, *Channels* contains their channel assignments.

Run *Cluster*() algorithm with inputs *UserPositions* and *NumberOfClusters* to cluster the data, obtaining the positions of the centroids of the clusters.

Set *RepeaterPositions* equal to the positions of the centroids.

Run *GetRepeaterRanges*() with inputs *UserRanges*, *RepeaterPositions*, and *MaximumRegionRange* to obtain *RepeaterRanges*

Run *ConnectComponents*() with inputs *RepeaterPositions*, *RepeaterRanges* and *MaximumRepeaterRange* to obtain new values for those variables. This connects any disconnected repeaters.

Obtain *RepeaterAdjacencies*, the adjacency matrix of the network by determining which repeaters are within range of each other.

Run *AssignChannels*() with input *RepeaterAdjacencies* to obtain *Channels*, which contains the channel assignment for each repeater.

Note: The $Cluster()$ algorithm referred to above is simply the k-means clustering method described earlier in this section.

4.5.1 Connecting Clusters

By using a clustering process we can obtain some given number of positions at which to place repeaters to extend coverage to users. While some of these repeaters might be within range of each other, it is unlikely that they form a network with a single component. It is far more likely, especially when working with the clumped population models discussed above, that there exist at least two disconnected components in our repeater network. As mentioned previously, we would like all users within range of a repeater to be able to communicate with any other user that is within range of a repeater. Thus, we must somehow connect the disconnected components of our network. Since our goal is to minimize the number of repeaters used, we would like to do so using as few repeaters as possible.

Let $d_{i,j}$ be the minimum distance between any node in component i and any node in component j. Consider the components of our network to be the nodes of a graph where the weight of the edge between component i and component j is $d_{i,j}$. In conceptual terms, the most efficient way to span this graph is by constructing a minimum spanning tree. Similarly, the shortest way to connect the components in our network is by constructing a minimum spanning tree using repeaters. This will ensure that all of our network components are connected while using as few repeaters to do so as possible.

We used Algorithm 3 (detailed in pseudocode below) to accomplish this goal.

Algorithm 3 Connect Components

Require: $RepeaterPositions$, $RegionRanges$, $MaxRegionRange$

Ensure: All of the repeaters in $RepeaterPositions$ are part of the same component of the repeater graph. Repeaters are connected if they are within range of each other.

while $NumberOfComponents > 1$ **do**

Let $minDist$ be the minimum distance between two repeaters in different components. Let $startRepeaterPosition$ and $endRepeaterPosition$ be the positions of these repeaters.

Let $numRepeatersRequired$ be the number of repeaters required to span $minDist$ if their ranges are $MaxRegionRange$.

Add a repeater $\frac{minDist}{numRepeatersRequired}$ miles away from $startRepeaterPosition$ in the direction of $endRepeaterPosition$. Note that this repeater will be part of the same network component as $startRepeater$.

end while

Note: This algorithm essentially constructs a minimum spanning tree of a graph containing the disconnected repeater components in our network as nodes. We then use that minimum spanning tree as part of our network to ensure continuity.

4.5.2 Assigning Channels

Once the network is connected, channels must be assigned to implement the interference-reducing structure described in Section 4.4.

First, we must choose a repeater to serve as the hub of the network. We prefer a hub which has the least average geodesic distance to other nodes in the network (also known as the greatest closeness centrality) for a number of reasons. First, this reduces the likelihood that we will need to include an extended chain (see section 4.2) in our network, which requires more nodes than a bi-directional chain. Also, choosing a hub with the least average geodesic distance minimizes the number of nodes that will be tied up by a message passing through the hub.

Once we have chosen a hub, we assign channels of increasing number stepping outward from the hub as described in Algorithm 4 below. Note that repeater frequencies are assigned according to the table below:

Channel Number	Frequencies
1	A
2	B/A′
3	C/B′
4	D/ C′
5	E/D′

Algorithm 4 Assign Channels

Require: *RepeaterAdjacencies*, the adjacency matrix for our repeater network.

Ensure: *Channels* contains a list of the channel assignments of each repeater such that each repeater is of the proper channel to communicate with its neighbor closest to the hub of the network.

Find the repeater with the highest closeness centrality in the network. This is our hub repeater. Assign it channel 1.

Let $CurrentChannel = 1$.

Let *ConnectedRepeaters* contain all of the repeaters connected to the hub.

while At least one repeater does not have a channel **do**

  Let *NextConnectedRepeaters* be an empty array

  for all Repeaters in *ConnectedRepeaters* **do**

    Assign the repeater to channel $CurrentChannel + 1$.

    Add all of the repeaters connected to the repeater that do not have channels to *NextConnectedRepeaters*

  end for

  Increment *CurrentChannel*

  Let *ConnectedRepeaters* equal *NextConnectedRepeaters*

end while

5 Results

5.1 Procedure

We evaluate our model by attempting to measure the overall quality of the network. The first metric we measure is the coverage of the network by looking at the number of people not covered the network. In addition to this, we generate results of our algorithm for different cluster sizes which effects the number of repeaters used in the network. The last metric we use to measure this is the transmission capacity for our network, which is very

# of clusters	Population Distribution				
	Random	5 clumps	10 clumps	20 clumps	40 clumps
5	17.00 ± 0.00	12.60 ± 4.98	16.60 ± 0.89	17.00 ± 0.00	17.00 ± 0.00
10	36.20 ± 1.10	24.60 ± 5.90	27.80 ± 3.63	26.60 ± 2.61	31.00 ± 4.24
15	43.40 ± 2.19	37.00 ± 5.10	39.00 ± 1.41	37.00 ± 2.83	40.60 ± 2.97
20	55.00 ± 2.45	49.00 ± 6.48	50.60 ± 2.97	50.60 ± 2.97	51.00 ± 3.16
25	74.60 ± 3.85	61.80 ± 4.82	62.20 ± 1.79	64.20 ± 3.35	65.80 ± 3.35
Naive	43.20 ± 0.84	25.2 ± 1.92	33.0 ± 2.45	33.6 ± 2.610	40.0 ± 1.58

Table 1: Mean number of repeaters that were used in constructing the network

important for the network to be usable. We measure this for both our algorithm and the naive algorithm to provide a comparison between the two. Sensitivity analysis is calculated for each of the metrics by determining the variability of each metric within a given population.

We generate clumped population distributions containing different number of clumps sizes (5, 10, 20, and 40) to test the network. A random population distribution is used as well to see how our network deals with a disperse population. We test the algorithm on this variety of different distributions to account for the variability of population distributions with different locations. For each of these 5 types of population distributions we generate a 5 test populations data set to allow us to measure the variability of our algorithm within a given population distribution.

5.2 Number of Repeaters

The number of repeaters used in the naive algorithm is simply determined by the number that it takes to completely cover the network. In our algorithm the number of repeaters initially placed is based on the number of clusters used in the k-means algorithm. More repeaters can also be added to connect disconnected regions of the graph. In general though, we see an increase in the number of repeaters being placed as the # of initial clusters increases.

5.3 Coverage

The naive algorithm is guaranteed to have 100% coverage by construction. It is guarenteed to cover all of the people regardless of how many repeaters

	Population Distribution				
# of clusters	Random	5 clumps	10 clumps	20 clumps	40 clumps
5	382.20 ± 25.11	18.40 ± 23.38	84.60 ± 56.18	183.0 ± 52.65	294.40 ± 51.93
10	88.80 ± 24.05	5.400 ± 1.14	8.80 ± 1.92	46.60 ± 11.50	50.60 ± 28.48
15	18.40 ± 8.17	5.800 ± 1.92	16.40 ± 20.49	17.60 ± 5.18	21.80 ± 4.09
20	10.20 ± 2.17	5.800 ± 2.17	7.80 ± 3.49	10.20 ± 2.28	12.20 ± 1.48
25	7.80 ± 1.64	6.00 ± 1.58	6.60 ± 0.89	10.80 ± 4.09	9.600 ± 2.61
Naive	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

Table 2: Mean number of people who were not covered by the network for different population distributions

that it must place. Our k-means clustering algorithm has a decrease in the population that is not covered by the network as the number of clusters used increased. This corresponds to more repeaters being placed in our preliminary construction of the network, which corresponds to a larger region being covered by our network.

5.4 Transmission Capacity

The transmission capacity is measured through a simulation of potential requests made across the network. This metric has the potential to be very variable for different requests so it is run 10 times to get an average value. We perform sensitivity analysis by measuring the standard deviation in the results across the 10 simulations. This is done for each population distribution to see if this metric remains effective across all distributions.

	Population Distribution				
	Random	5 clumps	10 clumps	20 clumps	40 clumps
K-means	0.0033	0.0144	0.0104	0.0104	0.0052
Naive	0.0065	0.0076	0.0083	0.0076	0.0075

Table 3: Standard deviation of the transmission capacity metric for different population distributions

The standard deviation of the transmission capacity was $\approx 1\%$ for most of the population distributions. For the naive algorithm the standard deviation for each one was approximately uniform illustrating that the metric

	Population Distribution				
# of clusters	Random	5 clumps	10 clumps	20 clumps	40 clumps
5	0.621 ± 0.022	0.715 ± 0.102	0.803 ± 0.047	0.780 ± 0.050	0.687 ± 0.033
10	0.906 ± 0.029	0.850 ± 0.046	0.951 ± 0.030	0.921 ± 0.019	0.943 ± 0.029
15	0.977 ± 0.010	0.830 ± 0.074	0.938 ± 0.058	0.967 ± 0.010	0.974 ± 0.004
20	0.984 ± 0.004	0.828 ± 0.073	0.950 ± 0.034	0.967 ± 0.014	0.981 ± 0.003
25	0.985 ± 0.003	0.814 ± 0.042	0.900 ± 0.037	0.954 ± 0.011	0.980 ± 0.004
Naive	0.221 ± 0.002	0.313 ± 0.016	0.274 ± 0.008	0.252 ± 0.010	0.236 ± 0.002

Table 4: Percentage of transmission requests that were processed by the clustering network and naive network for different population distributions

was valid across different population distributions. For our model using the k-means algorithm the standard deviation was a bit larger for the 5 clump data set of populations. However, even this had a standard deviation of $< 1.5\%$ which illustrates that our metric for transmission capacity was consistent.

The results for the transmission capacity show a significant improvement in our algorithm to process requests over the network in comparison to the naive approach. In addition to this, we notice that as we increase the number of initial clusters the transmission capacity of our network increases. This creates a trade off between the number of repeaters and the transmission capacity and allows a balance to be chosen between the two based on the purpose of the network.

6 Discussion

6.1 Effects of 10,000 Users

In addition to developing a method for determining the minimum number of repeaters necessary to accommodate 1000 simultaneous users, we were also asked how our solution would change if there were instead 10000 users.

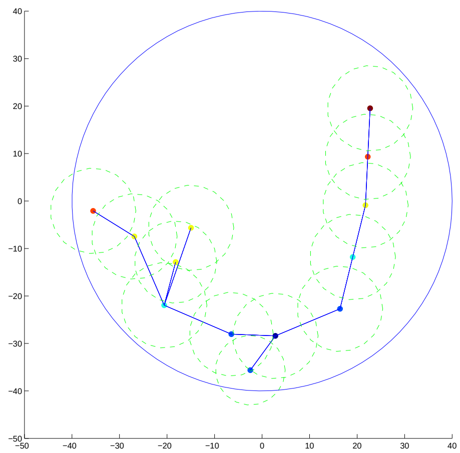
Our algorithm can easily be run to generate networks to serve 10000 users. However, the quality of service these networks provide is uninspiring. In general, our algorithm's networks were only able to successfully handle about half of the 10,000 transmissions attempted over 200 units of time for any number of repeaters between 15 and 20. While this is significantly better than the naive algorithm which only handles around 700 of 10,000 requests, it is obviously not ideal.

Increasing the number of users by an order of magnitude increases network traffic by an order of magnitude as well, at least according to our assumptions. As a result, each repeater is roughly ten times as likely to be tied up in each timestep. This leads to many transmissions being blocked, especially transmissions through the hub of the network.

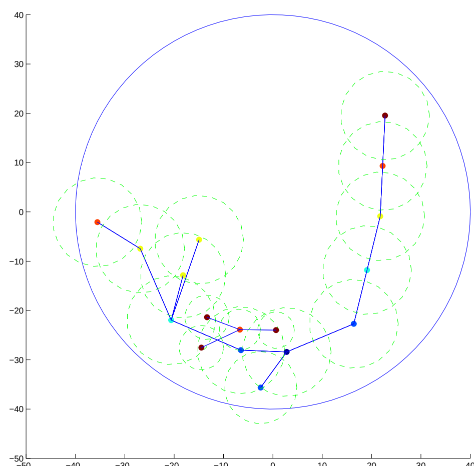
We have developed a number of approaches to dealing with this problem. Our modeling of user transmissions indicates that approximately one third of user communications are between users that are only one repeater apart (i.e. within range of the same repeater). These sorts of communications tie up that repeater, blocking transmissions through that part of the network. If such communications could be routed in a way that did not block the repeater, it follows that the network's capacity would increase significantly.

One strategy that we have considered adding to our algorithm is the ability to recursively add subnetworks to our network (using the same process as our main algorithm) at nodes where user density and therefore demand is high. These subnetworks are composed of repeaters with frequencies incompatible with the frequencies of the adjacent main node, eliminating the possibility of intereference. (see figure 4 below). Unfortunately, time constraints have prevented us from finishing this feature or conducting an analysis of its effects.

Even though the idea hasn't been fully developed, we illustrate how a subnetwork would look in the figure below. The addition of the 3 repeaters occurs near the central hub, which, in this example, we are assuming is an area of high population density. Therefore by adding this subnetwork, we would be assuming that the benefit added by increasing the transmission capacity of the network would outweigh the cost of adding the extra repeaters. A nice feature of this structure is it would allow you to place cost parameters on the repeaters and the transmission capacity to determine in what situations that it would be beneficial to place an extra repeater.



(a) No subnetwork



(b) With subnetwork

Figure 5: A typical network generated by our algorithm with a localized subnetwork included

One of the weaknesses of our network design is its dependence on a central hub, which in many cases serves as a chokepoint limiting the flow of cross-network communications. If more private lines were available for use we could use them to add multiple hubs to our network, similar to the way IP addresses are used to route traffic to different hubs on the Internet.

6.2 Effects of Mountainous Terrain

Currently our algorithm assumes a flat terrain devoid of features that might impact the range of repeaters or mobile devices. However, we feel that our algorithm could be more or less readily adapted to deal with defects in line-of-sight-propagation caused by mountainous areas.

The main complication introduced by mountainous areas is the introduction of non-symmetrical maximum repeater ranges that vary with position. This would mainly affect the way in which we use clustering to place our initial repeaters and our method for connecting the disconnected components of our network. The other parts of our algorithm, specifically the use of private lines and the assignment of channels, are dependent only on the adjacencies of repeaters in our network and would not be affected.

While more experimentation is needed, we feel that the use of clustering would still serve well as an initial method for determining where to place repeaters. However, we would then attempt to optimize the position of the repeaters to encompass more users through an optimization technique such as simulated annealing.

A similar approach could be taken to connect disconnected network components. The straight-line minimum spanning tree approach could be used as a starting point for the placement of additional repeaters, with optimization then applied to try to maximize the reduction in the distance between disconnected components.

Obviously, this additional optimization would come with significant runtime penalties.

We feel that mountainous terrain could even decrease the number of repeaters needed, since repeaters could be placed on elevated terrain to achieve increased range.

7 Conclusion

Overall, the proposed algorithm tends to do a pretty good job of producing a network that passes the “eyeball test” from a heuristic viewpoint. That is, our algorithm proposes a solution that mimics the overall shape of the population distribution, covers the vast majority of potential users and uses few obviously wasteful repeater nodes.

While the algorithm performs well in many situations, when the number of clusters used to generate our initial repeaters is too large, we run into problems. Our network is heavily dependent upon using privacy lines to control the flow of signal traffic. Because there are a limited number of privacy lines available to us, this puts a hard cap - of having repeaters in at most about 25 distinct locations - on the number of repeaters that can be used. As a result, there is also an effective cap on the number of initial clusters that we can use to generate our initial repeater locations. Once we get past about 20, we no longer have enough privacy lines to implement the signal-control scheme outlined in section 4.4.

Though the cap on the number of initial clusters reduces our options when it comes to designing a network, it also greatly diminishes our problem space. Because of the limited number of options, we could very easily search across all possible numbers of initial clusters (in increasing order) to find the minimum value that gave us a network satisfying our transmissions capacity requirements for a given population distribution. This approach is not as direct as searching across the number of available nodes, but it should give us the best possible solution generated by our approach for a particular population distribution.

When our algorithm does receive a good value for the number of clusters in the population, the data shows that the network it creates does an excellent job of handling traffic under the assumptions that we have made. For example, for the case of 15 initial clusters above, the algorithm created a network that covers over 97% of the population and is able to transmit $\approx$ 95% of the requests on average. This performance corresponds to a network that typically contains only about 40 repeaters. Overall, when used properly, our algorithm can produce very efficient, effective networks. We feel that it could serve as a useful tool in VHF radio network planning.

8 Future Work

8.1 Limitations of our model

Though we feel our algorithm is quite strong in general, it does have a number of limitations. For one thing, *k*-means clustering is a non-deterministic algorithm. It simply finds a "good" way to cluster the data and runs with it. Because of this, for a given population distribution, our algorithm doesn't return the same result each time. This isn't the most reassuring when we're supposed to be finding the "optimal" solution.

In addition to the concern above, the method of using privacy lines for controlling the flow of where a signal gets broadcasted contains a potential complication in certain circumstances (recall figure 3 above) . In real-world applications, repeaters take a bit of time to process a signal before re-broadcasting it. When both the 5/11 and 4/12 towers pick up the signal from 6/13, there's a very real chance that the two repeaters will process the signal at a slightly different speed. If this happens, then 1 would receive two identical signals, slightly offset, which would be less than ideal. Using the same brand of repeater throughout the network might help mitigate the impact of this issue. In addition, with enough private lines (though admittedly, you'd need a lot), this effect can be completely addressed.

The limited number of private lines imposes a couple of very important limitations on our model. First of all, it introduces a bound on the number of repeaters that can be used in the overall network. Because each pair of repeaters needs two distinct private lines and private lines must be used to specify both local communication and communication across the entire network, the overall network can have at most 25 distinct repeater locations. In addition, we have limited tools to limit the amount of demand on key repeaters such as the hub. As mentioned above, we use privacy lines the way the internet uses IP addresses. With a larger number of them at our disposal, we could create a larger (if needed), more flexible network that could route much of its flow around the hub.

The algorithm as currently implemented can really only be used for optimizing with respect to the number of repeaters. Someone who is considering this problem may wish to, for example, enforce a steep penalty for not including a user in the network. Presently, the model doesn't allow for this. More generally, the model doesn't allow the user to weigh the cost of including an extra repeater versus the other benefits that extra repeater would bring.

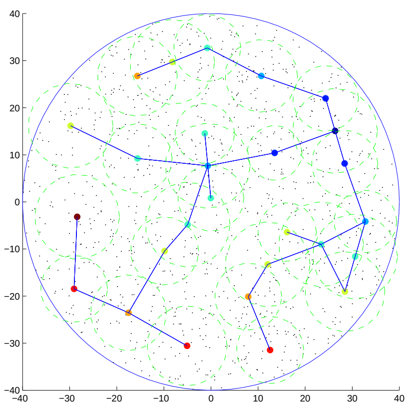
Another area that the model fails to address is any kind of variation in atmospheric conditions or weather that could reduce the broadcast range of our repeaters. This is a much larger concern for chains of repeaters, because if the range of any is sufficiently reduced it could break the chain. Though it is not explicitly addressed, the risk of this concern can be mitigated by building a buffer into the maximum range of our broadcast towers (for example, assuming a 7-8 mile broadcast radius when we actually have a 9-mile radius).

Finally, the code currently doesn't allow the user to exactly specify the number of repeaters being used for a given network—rather, this is roughly controlled by the number of clusters given as input to the k-means clustering algorithm. The ability to exactly specify could relatively easily added via a post-processing step in which the repeaters whose removal would lead to the least reduction in network quality are removed until the desired number of repeaters is reached.

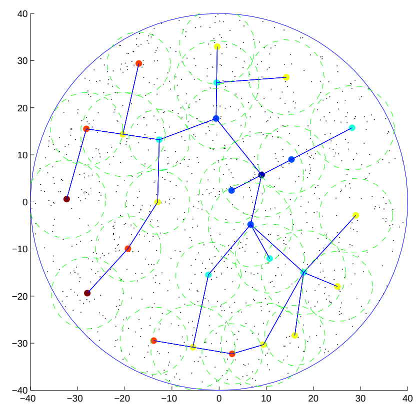
8.2 Ways to improve the model

In addition to the issues raised in our "Discussion" section, there were several other considerations that we did not have time to fully address. First, we could extend our model to take into account the "blind interference" phenomenon discussed in the assumptions section. Second, in the real world, repeaters do not immediately transmit radio messages; ideally we would take into account this nontrivial switching time in a more rigorous fashion. Third, we assumed throughout that users make equal demand on the system, something that just isn't true in the real world. Including the option for a user to have higher demand than others - something that could be simulated in our model by placing "multiple users" at the location of a heavy user—would increase the value and accuracy of the model. Finally, it would be nice if our heavily-stressed hub repeater was located in a less populous area to isolate it from local demand.

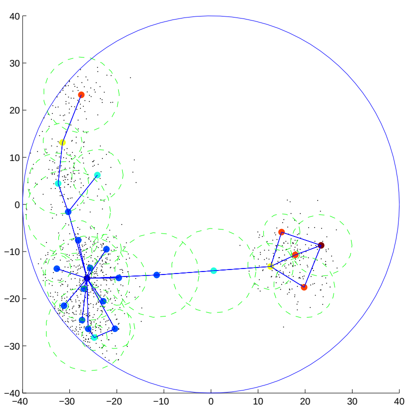
A Appendix



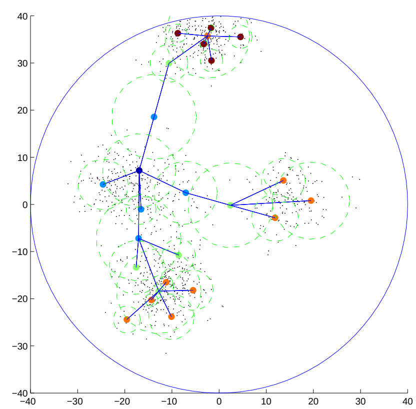
(a) Random Population Distribution



(b) Random Population Distribution

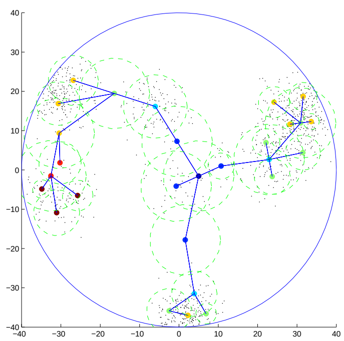


(c) Clumped Distribution - 5 Clumps

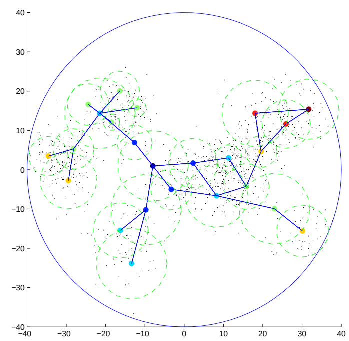


(d) Clumped Distribution - 5 Clumps

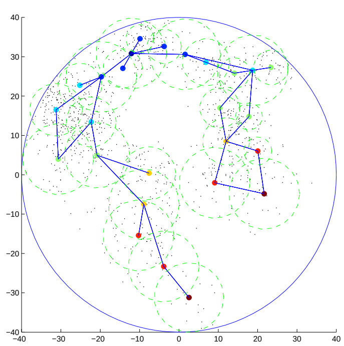
Figure 6: Examples of the network that our algorithm constructs on various population distributions



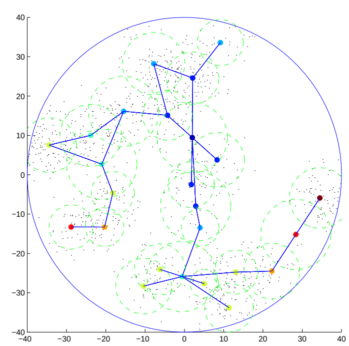
(a) Clumped Distribution - 10 Clumps



(b) Clumped Distribution - 10 Clumps

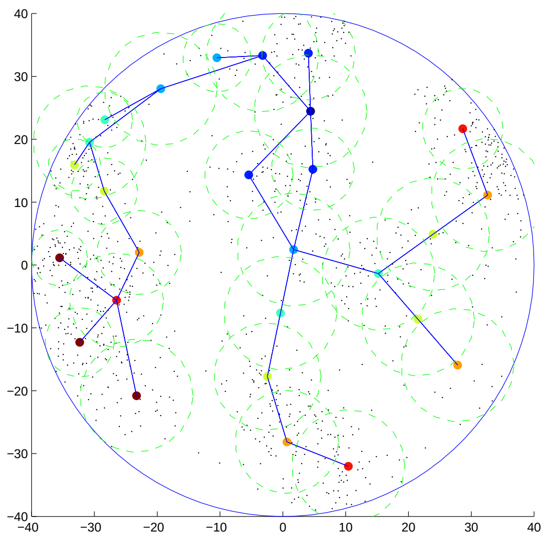


(c) Clumped Distribution - 20 Clumps

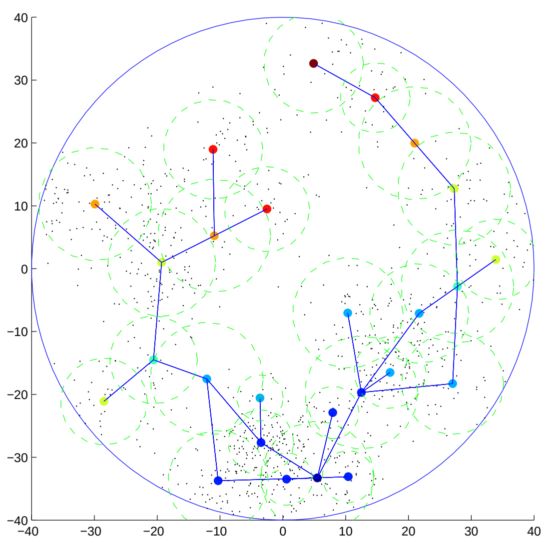


(d) Clumped Distribution - 20 Clumps

Figure 7: More examples of the network that our algorithm constructs on various population distributions



(a) Clumped Distribution - 40 Clumps



(b) Clumped Distribution - 40 Clumps

Figure 8: A few examples of the network that our algorithm constructs on various population distributions

References

[1] Socioeconomic data and applications center. "http://sedac.ciesin.columbia.edu/gpw/". Viewed on 2011 February 13.

[2] Vhf radios: Why you need one. "www.boatus.com/boattech/vhf.htm", 2007. Viewed on 2011 February 12.

[3] Information for the amateur radio enthusiast. "www.hamdata.com/fccinfo.html", 2011. Viewed on 2011 February 11.

[4] The Pacific Crest Company. VHF/UHF Range Calculations. "http://www.mertind.com/argentina/Soporte20tecnico/Instrumentos/PDL/PDL20UHF-VHF20RANGE20CALCULATIONS.pdf". Viewed on 2011 February 11.

[5] Otis Dudley Duncan. The measurement of population distribution. *Population Studies*, 11(1):pp. 27–45, 1957.

[6] The American Radio Relay League. Us amateur radio frequency allocations. "www.arrl.org/frequency-allocations". Viewed on 2011 February 11.

[7] David MacKay. Information theory, inference and learning algorithms, chapter 20. "http://www.inference.phy.cam.ac.uk/mackay/itprnn/ps/284.292.pdf", 2003. Viewed on 2011 February 11.

[8] The National Radio Astronomy Observatory. How radio waves are produced. "http://www.nrao.edu/index.php/learn/radioastronomy/radiowaves", 2002. Viewed on 2011 February 11.

[9] John Owen. VHF/UHF TV Line of Sight Calculator. "http://www.naval.com/sight/index.htm", 2003. Viewed on 2011 February 11.

[10] The International Amateur Radio Union. Ethics and operating procedures. "http://www.hamradiosecrets.com/support-files/ham-radio-operator-guidelines.pdf", 2009. Viewed on 2011 February 11.