# First Steps in Mesh WiFi Network Design by Genetic Algorithm

**M.B. Reed\* and S. Nash**

*Department of Mathematical Sciences, University of Bath, Bath, England, U.K., \*Email: m.b.reed@bath.ac.uk.*

**ABSTRACT:** Given a mesh of wireless nodes for WiFi customers covering a city district, we describe a genetic algorithm-based approach to the problem of selecting a small fixed number of nodes as gateways to the internet, and linking the remaining nodes to the gateways either directly or by 'hopping', to create an efficient mesh network structure. The algorithm uses a modification of *k*-means clustering to allocate nodes to gateways.

KEYWORDS: Genetic algorithms, Mesh WiFi network.

**الخطوات الأولية في تصميم شبكة واي-فاي (WiFi) متشابكة بواسطة خوارزمية وراثية**

**مارتن ريد و ستيوارت ناش**

**ملخص:** بافتراض وجود شبكة نقاط لاسلكية لمستخدمي شبكة واي-فاي (WiFi) تغطي منطقة في مدينة ما، نصف خوارزمية وراثية لمسألة اختيار عدد ثابت وصغير من نقاط الشبكة كبوابات إخراج إلى الشبكة العنكبوتية مع ربط باقي نقاط الشبكة إما بطريقة مباشرة أو "بالقفز" لتكوين هيكلية فعالة للشبكة. تستخدم الخوارزمية تعديلا لمتوسطات k تجمع وذلك لتحديد نقاط بوابات الخروج.

## 1. Introduction

A Mesh WiFi network is a city-wide wireless network providing broadband connections for consumer equipment such as laptops and smartphones. Coverage is provided by WiFi nodes attached to masts, lamp-posts and other street furniture. Connecting each of these nodes directly to the wired internet would be prohibitively expensive, so a mesh network is formed, where only a small proportion of nodes are directly connected (these are called gateway nodes), and the remaining nodes transmit their data to a gateway, either directly or by 'hopping' via one or more intermediate nodes. We are given the nodes' locations, and a list of which nodes have the potential for being gateways, as in Figure 1. We also know, for each node, a list of those other nodes which it can 'see', i.e. would be able to transmit to. These links are limited by distance and also by line-of-sight; two nearby nodes may not be able to connect to each other because of an intervening building. The initial graph contains the potential connections as edges; see for example Figure 2. The task is to design an efficient mesh structure by identifying a small number of nodes which will be used as gateways, and the route by which each of the other nodes will link to a gateway. An efficient network is one which is feasible (each node is connected to a gateway, directly or indirectly, using valid links), and in which:

- The number of gateways $G$ is relatively small;
- Each gateway services approximately the same number of nodes, i.e. the load or bandwidth is balanced among gateways;
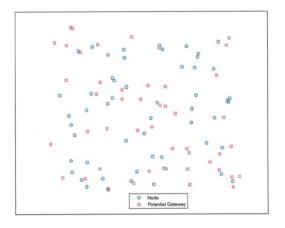- The amount of data-hopping from nodes to gateways is minimised.
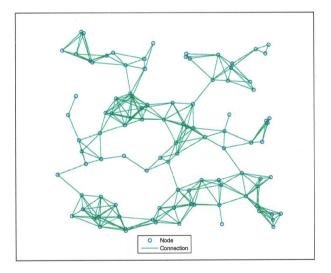


Figure 1. Example 1: Node locations.



Figure 2. Example 1: All possible connections.

A search of the literature - in particular of the relevant Institute of Electrical and Electronic Engineers (IEEE) conferences - has not revealed any published algorithmic approaches to this problem. As will be seen in Section 4 below, it is straightforward to evaluate a cost function for a given Mesh WiFi network structure. However, the structure itself is complex, consisting of a division of the initial graph into $G$ separate subgraphs, each having one of its nodes designated as a gateway and (direct or indirect) connections from the other nodes to the gateway node. Two general approaches to algorithm design for this combinatorial problem might be:

    i.    First assign the nodes to $G$ clusters, then within each cluster select the gateway node and hence choose

the edges needed to connect the other nodes to it; or

ii.   First select the $G$ gateway nodes, then 'grow' the subgraphs from each gateway until all nodes are included in a subgraph.

A combinatorial optimization problem with characteristics similar to the Mesh WiFi problem, is the Vehicle Routing Problem (VRP). In a VRP, a fleet of vehicles must be routed to deliver goods from a central depot to a distributed set of customers. The VRP thus involves the interdependent tasks of associating each vehicle with a cluster of customers, and for each vehicle determining the most efficient route by which the vehicle can visit those customers, all routes beginning and ending at a central depot. The customers are treated as nodes on a graph, and we seek a shortest-path route for each vehicle such that each customer node is visited by exactly one vehicle. Here we also find two general categories of heuristic constructive algorithms which have been proposed in the literature, namely 'cluster-first, route-second' and 'route-first, cluster-second'. For example Beasley (1983) gives an example of a route-first cluster-second algorithm. This categorisation of algorithms also applies to the more complicated Capacitated Arc Routing Problem (CARP), where the customers are distributed along arcs of a graph, and each customer has a demand quantity $wj$; the total demand on a given route must not exceed the vehicle capacity $W$. In the first approach, the graph is partitioned into clusters containing adjoining arcs, each cluster having a total demand not exceeding $W$, using a greedy algorithm or general assignment algorithm; then for each cluster a shortest-path route around the arcs is required (a standard problem in Operations Research known as the Chinese Postman Problem). The second approach may involve first constructing an Euler 'grand tour' over all the arcs, which is then partitioned into feasible sub-tours, each satisfying the capacity constraint. Eiselt *et al.* (1995) review published algorithms for the CARP. Practical applications include the routing of street sweeping and road-gritting, postal delivery and waste and recycling collection.

The CARP is an NP-hard (non-deterministic polynomial-time hard) problem. When comparing it to the Mesh WiFi problem, however, we see an additional level of complication in the latter: within each cluster we do not have to find a route around the arcs/nodes starting and finishing at a central depot node, but we need to efficiently link the nodes to a gateway node which must itself be chosen from among the cluster nodes as part of the problem. The CARP can be extended to the Multiple Depot VRP. In the MDVRP a fixed number of depot locations are prescribed. Practical applications include consumer goods distribution and newspaper delivery. A recent paper by Ho *et al.* (2008) approaches the MDVRP using genetic algorithms.

If the optimal location of the depots (and possibly also the optimal number of depots) must be determined as part of the problem, we have the Location Routing Problem (LRP). The LRP is an extension of the MDVRP in which a (prescribed or optimal) number of depot facility sites must be chosen from a set of potential depot locations, and a cluster of customers built around each facility, which can then be routed. This is the routing problem closest to the Mesh WiFi problem. The LRP comprises a Facility Location Problem and a VRP as sub-problems, each of which is NP-hard. A review of LRP algorithms by Tuzun and Burke (1999) suggests that the most profitable approach to this problem is an integrated or two-phase algorithm, which iterates between the location and routing sub-problems. A practical LRP problem solved numerically involved location of regional blood-banks serving hospitals in a region.

A recent development in solving combinatorial optimization problems is the use of metaheuristics, which provide a general framework for directed search within the set of feasible solutions. Under this heading we may mention simulated annealing, tabu search, neural networks, and evolutionary computation methods. In the last category the most well-established metaheuristic is genetic algorithms, although the power of ant colony optimisation (Dorigo and Stuetzle, 2004) is becoming increasingly recognized. This paper proposes a genetic algorithm (GA) based method for the Mesh WiFi problem. An individual in the GA is defined just by the unordered list of its gateway nodes; from these, the allocation of nodes to gateways is done by a clustering algorithm. The clustering algorithm is also used to create the initial population of feasible solutions. Very simple crossover and mutation operators are used; crossover is performed essentially by picking low-cost gateways from the two parent solutions. The GA was run for 50 generations. It was tested on randomly-generated mesh layouts of up to 100 nodes.

## 2.  Genetic algorithms

Genetic algorithms were first proposed by Holland (1975), and popularised in the textbook by Goldberg (1997). They are used to find single or multiple optima of a fitness function $F$. They are zero-derivative methods, and require no smoothness conditions on $F$. They work with an encoding of feasible solutions, for example as binary strings, ordered or unordered lists of letters or integers, or real-valued vectors. Each encoded solution is called an individual, or chromosome, and its individual elements are genes. The GA algorithm starts from an initial population of $N$ individuals, usually randomly-generated. The fitness of each individual is evaluated. The GA then proceeds to develop a next-generation population by first copying all the current population. It then augments and refines this population using combination, mutation and selection, as follows:

   i.   **Combination:** Two parent individuals are selected by weighted roulette wheel selection (individuals of higher fitness have a greater chance of being selected), and combined to produce one or more offspring. The possible combination operations usually include a variant of the biological process of genetic crossover;

   ii.   **Mutation:** The fittest of the offspring is chosen, but before being added to the new population a process of random mutation is applied to its genes; typically each gene has a 0.01 chance of being mutated (e.g. by flipping between 0 and 1 in a binary string). The combination+mutation process is repeated $k$ times, creating $k$ new population members;

   iii.   **Selection:** The new, augmented population of size $N + k$ is reduced to size $N$ by a 'survival of the fittest' process.

The above procedure is repeated with the new population, and iterated until a population containing individuals of sufficiently high fitness is obtained. In line with the Simple Genetic Algorithm (SGA) of Goldberg, we create two offsprings from each combination using crossover, and perform the selection process using binary tournament: in each tournament, two individuals are picked at random, and the individual with the lower fitness is marked for deletion.

As with the Location Routing Problem discussed above, the Mesh WiFi problem is not amenable to a standard implementation of GA. It involves two sub-problems: the Gateway Location Problem (GLP) and the Mesh Routing Problem (MRP), both of which are NP-hard, and which are interdependent. The fitness of a selected set of gateways cannot be assessed without forming a mesh routing solution, and mesh routing can only be performed on the basis of a set of gateways. A complete feasible solution, prescribing the gateway locations and the routing, is too complicated a phenotype to be expressible as a genotype such as a binary string. Our solution to this problem is to solve the GLP using a genetic algorithm, in which the chromosome is just an unordered list of gateway nodes. From such a GLP solution a MRP solution is grown using a clustering algorithm.

The following three sections describe features of the GA applied to the Mesh WiFi problem. We denote the number of nodes by $N$, and the number of gateways by $G$; typically $0.1N < G < 0.2N$.

## 3.  Features I: Encoding and the clustering algorithm

The success of a GA for a particular situation depends greatly on the way that solutions are encoded. A full encoding of a mesh network could be done using a vector of $N$ integers, where the $i$th integer either indicates that node $i$ is a gateway node (by using a value of 0, for example), or gives the number of the node to which node $i$ will transmit its data. However such an encoding does not reflect the geometric properties of the network; the primary property is the allocation of nodes to gateways, i.e. the partition of the $N$ nodes into $G$ clusters, each cluster containing a gateway node and the nodes which transmit to it either directly or indirectly. We therefore chose a simpler partial encoding which just contained the $G$ gateway node numbers, in an unordered list. From this list, the full network would be constructed using a clustering algorithm.

The principle of $k$-means clustering (Lloyd, 1982) is to partition a set of $N$ data-points into $k$ clusters, such that each point is in the cluster whose centroid is closest to it. The clustering process is usually performed by an

iterative algorithm (Lloyd's algorithm) involving two steps, starting from an initial partition whose cluster centroids have been calculated:

  i.   **Assignment:** Assign each point to the cluster with the closest centroid;
  ii.  **Update:** Re-calculate the centroids of the clusters.

The iteration continues until no re-allocation of nodes is occurring. While the general principle of this algorithm is relevant to the Mesh Wifi problem, there is a difficulty. As was already observed, two nodes may be close together spatially, but unable to link due to an intervening building. The clustering should thus be done on the basis of the potential connections, rather than spatial distances. In our variant of $k$-means clustering, the distance from node $i$ to the centroid of cluster $m$ was replaced by the proportion

$$p_{im} = \frac{v_{im}}{s_m}$$

where $v_{im}$ is the number of nodes in cluster $m$ which node $i$ can 'see' (i.e. is connected by an arc on the initial graph), and $s_m$ is the number of nodes in cluster $m$. In the iterative algorithm, node $i$ is assigned to the cluster with the highest value of $p_{im}$. The gateway nodes themselves do not get reassigned.

Once the iteration has converged, the network connections within each cluster are formed. First, all nodes in the cluster which can see the gateway node, are routed directly to it. Then all unconnected nodes which can see an already-connected node, are routed to that node - and so on, until all nodes in the cluster are routed by pathways to the gateway. This approach is also used to grow the initial clusters from the gateways, at the start of the k-means clustering algorithm.

The initial population of the GA consists of solutions formed by choosing $G$ gateways at random from among the potential gateway nodes, and then applying the clustering algorithm.

## 4. Features II: Fitness function

The fitness function used was

$$F = \frac{1}{1+C}$$

where the cost function $C$ is the sum of the costs for each gateway cluster. This is given by

$$C = \sum_{m=1}^{G} \max(0, c - c_m).$$

Here, $c$ is the average nodal bandwidth, defined as the total bandwidth $B$ for the network divided by $N$, and the cost $c_m$ of the $m$th cluster is

$$c_m = \frac{cN}{G(r_m+1)(s_m+1)} \tag{1}$$

where cluster $m$ has $r_m$ nodes connected directly to its gateway, and $s_m$ nodes connected indirectly (through hopping).

It can be checked that $C = 0$ in the ideal network of equal numbers of nodes in each cluster, and no hopping. The optimal fitness is thus $F = 1$, and the range is $0 < F \le 1$.

## 5. Features III: Genetic operations

In the Combination step of the GA, a pair of offsprings are formed from two parent solutions (a 'mother' and a 'father'), chosen by roulette-wheel selection. Each parent consists of an unordered list of $G$ gateway nodes. The list of gateways defining the first offspring ('child 1') is formed by first including all gateways which appear in both the mother and the father. Then we look at the cluster costs (as defined in the fitness function) of the

remaining gateways in the mother, and include the maternal gateway associated with the least cost. Any remaining gateways in child 1 are obtained from the father, again using the lowest-cost principle. Child 2 is formed in the same way but using the least-cost paternal gateway, and filling with low-cost gateways from the mother. The fitness of each child is then calculated. Note that this operation may produce infeasible solutions as offsprings if the initial graph is not fully connected.

Mutation is applied to the fitter offspring before it is added to the augmented population. Each gateway node is considered in turn, and has a 0.03 probability of being mutated. Where a mutation occurs, the node is replaced by a node which it can see, selected randomly. After mutation, the clustering is re-performed with the new gateway nodes.



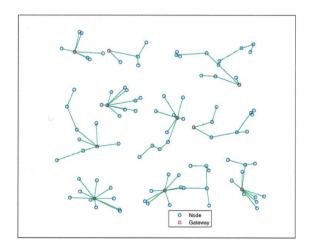Figure 3. Example 1: 10 Gateways: Best initial connections.



Figure 4. Example 1: 10 Gateways: Best final connections.

## 6. Results

The algorithm is in the early stages of development, but has already produced encouraging results on small

sets of artificial data. To generate the data, one hundred nodes were first positioned randomly over a unit square, and each of these had a 0.5 probability of being identified as a potential gateway (see Figure 1). Potential connections were also assigned with a random probability, between nodes sufficiently close together (Figure 2). An initial population of 50 solutions was then created by randomly selecting 10 gateways, and growing the clusters as described in Section 3. The fitness of each solution was calculated, and Figure 3 shows the highest-fitness solution in the initial population. The GA was then run for 50 iterations, and Figure 4 shows the best solution in the final population. A more even distribution of nodes into clusters is observed in Figure 4, although the cost has only decreased from 4.91 to 4.55. Figures 5 and 6 show the best initial and best final structures when 17 gateways are allowed. Here the cost has decreased from 2.00 to 1.27. In the second example, Figures 7 and 8, two exclusion zones were imposed in the domain, to represent adjoining buildings which prevent line-of-sight communication between nodes. In this case 17 gateways were specified, and the best initial and final structures are shown in Figures 9 and 10.
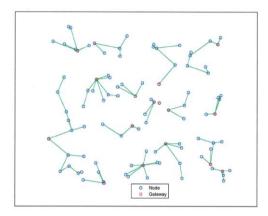


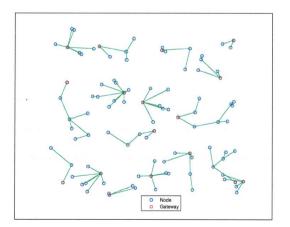Figure 5. Example 1: 17 Gateways: Best initial connections.



Figure 6. Example 1: 17 Gateways: Best final connections.

## 7. Conclusions

Initial results indicate that a genetic algorithm, utilising clustering, is a possible technique for assisting the

design of a Mesh WiFi network. Improvements could be made in many of the features of the current algorithm. Techniques from graph theory could be applied, in particular the Dijkstra algorithm for determining the shortest-path route between two given nodes, and the matrix-based algorithm finding the matrix of shortest path lengths between node pairs (Chen, 2003). This information could be used to check that the initial graph is fully connected, and to construct higher-fitness solutions for the initial population of the GA. The following are some further possibilities:
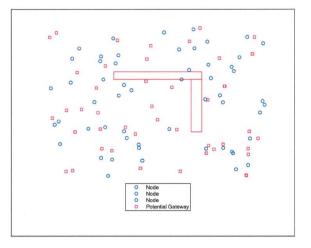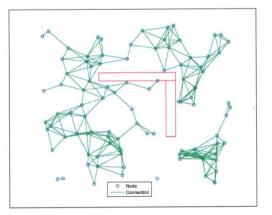


Figure 7. Example 2: Node locations.



Figure 8. Example 2: All possible connections.

i.    The clustering algorithm may be improved by using a different connection-based metric. For example, a node may be associated with the cluster for which the average path length to the other nodes in the cluster is smallest.

ii.    The fitness function is insensitive to poor internal construction of a cluster, when the size of the cluster is lower than average; there is thus no incentive for the GA to find the best node to act as the gateway. There should also be greater penalizing of paths involving more than one hop to reach the gateway, for example by replacing (1) by

$$c_m = \frac{cN}{G(r_m+1)(s_m+1)(t_m+1)^2}$$

where $s_m$ nodes need one hop, and $t_m$ nodes need two or more hops to reach the gateway.

iii. From looking at the initial graph showing all possible connections, it is clear that some nodes are more suitable as gateways than others, because of their high number of connections; this information could be used in the algorithm, by computing the degree of each node. The crossover and mutation operations, and the initial clusterings could be improved using this information.

iv. Once a mesh routing has been grown from a given set of gateways, the selection of the gateway node within each cluster could be optimized, by choosing the node with the lowest maximum path length to other nodes in the cluster, before redrawing the routes as described in Section 3.
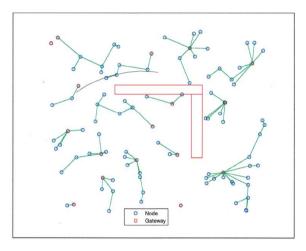


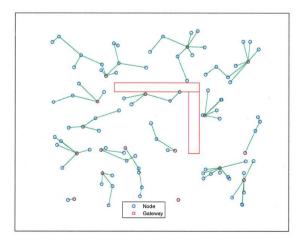Figure 9. Example 2: 17 Gateways: Best initial connections.



Figure 10: Example 2: 17 Gateways: Best final connections.

A further development would be to treat the number of gateways $G$ as a variable to be minimised. Following the practice with VRPs, this could be achieved by starting with a small value for $G$ and re-running the algorithm with increasing values of $G$ until feasible solutions of acceptable cost are found.

## 8.  Acknowledgements

## 9.  References

BEASLEY, J.E. 1983. Route First-Cluster Second Methods for Vehicle Routing. *Omega*, **11**:403-408.

CHEN, W-K. 2003. *Net Theory and its Applications: Flows in Networks*. Imperial College Press, London, UK.

DORIGO, M. and STUETZLE, T, 2004. *Ant Colony Optimization*. MIT Press, Cambridge, Massachusetts, USA.

EISELT, H.A., GENDREAU, M. and LAPORTE, G. 1995. Arc-routing Problems, Part II: The Rural Postman Problem. *Operations Research*, **43**(3): 399-414.

GOLDBERG, D.E. 1997. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley.

HO, W., HO, G.T.S., JI, P. and LAU, H.C.W. 2008. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, **21**(4):548-557.

HOLLAND, J. 1975. *Adaption in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor.

LLOYD, S.P. 1982. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, **28**:129-137.

NASH, S. 2007. *Motorola: Mesh Wifi Gateway Optimisation*. M.Sc. Dissertation, Dept. of Mathematical Sciences, University of Bath, UK.

TUZUN, D. and BURKE, L.I. 1999. A Two-phase Tabu Search Approach to the Location Routing Problem. *European Journal of Operations Research*, **116**(1):87-99.