Solving the Wireless Mesh Network Design Problem using Genetic Algorithm and Tabu Search Optimization Methods

Moheb R. Girgis
Professor, Department of Computer Science
Minia University
Minia, Egypt

Tarek M. Mahmoud
Professor, Department of Computer Science
Minia University
Minia, Egypt

Bahgat A. Abdullatif
Lecturer, Department of Computer Science
Minia University
Minia, Egypt

Ahmed M. Rabie
Minia University
Minia, Egypt

***Abstract* -Wireless mesh network (WMN) consists of mesh clients, mesh routers and gateways. In WMN, gateways connect to the Internet via wireline links and provide Internet access services for users. Due to the limited wireless channel bit rate, multiple gateways are usually required in a WMN, which costs budget and takes time to set up. WMN is a promising technology that provides wireless broadband access to end users. It offers a high degree of flexibility compared to traditional networks; however, this attribute comes at the expense of a more complex structure. Therefore, planning and optimization of WMNs are a challenge. In this paper, we address this challenge using a genetic algorithm and tabu search. The genetic algorithm and tabu search enable searching for a low-cost WMN configuration with constraints and determine the number of used gateways. Experimental results prove the effectiveness of the genetic algorithm and tabu search in minimizing WMN network costs while satisfying quality of service. The proposed models are shown to significantly outperform existing solutions.**

Keywords-Wireless mesh networks; genetic algorithms; tabu search; topology design.

I. INTRODUCTION

Wireless mesh networks (WMNs) [1] have become an important networking infrastructure because of their low cost and increased high-speed wireless Internet connectivity. A WMN has two types of nodes: mesh routers and mesh clients. Mesh routers are similar to conventional routers but incorporate additional functions to support mesh networking. They are typically equipped with multiple interfaces to accommodate different wireless technologies. Another feature that distinguishes mesh routers from conventional routers is their ability to provide the same coverage through multi-hop communications with much less transmitter power. In addition, mesh routers can be installed on dedicated or general-purpose machines. On the other hand, mesh clients are equipped with functions necessary for mesh networking and can also serve as routers; however, they are unable to function as gateways or bridges. Their single wireless interface with hardware and software platforms is much simpler than the mesh router interface. WMNs are based on a mesh topology in which every node (representing a server) is connected to one or more nodes, thereby enabling information transmission in more than one path. Path redundancy is a robust feature of this topology type. Compared to other topologies, mesh topology does not require a central node. This attribute enables networks of this topology type to be self-healing. Consequently, these types of networks are reliable and robust to potential server node failures.

On account of the above characteristics, rapid development of WMNs has been further compelled by their low associated costs, such as by avoiding the expense of deploying and maintaining wired Internet infrastructures. This makes WMNs an economical alternative for providing wireless Internet connectivity, especially in developing countries. Applications of WMNs include those for urban areas; metropolitan area networks; municipal wireless mesh networks; corporate and enterprise networks; community, neighbourhood, and home networks; medical, transport, and surveillance systems; building automation; among others [2]. Across these applications, WMNs provide cost-efficient broadband wireless Internet connectivity to end users. Several optimization problems have demonstrated their applicability to the efficient design of WMNs. These problems relate to optimizing network connectivity, user coverage, and stability, among other aspects. Their resolution is crucial for optimizing network performance [3]. In this paper, we propose and evaluate genetic algorithms (GAs) [4,5] and tabu search (TS) [6] for near-optimally solving one of these problems—minimizing costs—as it relates to WMN design.

GAs are evolutionary algorithms that are intended to implement the selection process as it occurs in nature. GAs begin from an initial population of 'individuals'—i.e., feasible solutions of a problem—with each solution having an associated fitness value that indicates how fit it is compared to the others. Thus, as in nature, where there are natural processes of selection, reproduction, and mutation, a GA experiences a similar process of evaluation, selection, crossover, mutation, and replacement, thereby engendering the next generation of individuals. The process is repeated through a number of generations during which the best features of parents are passed to offspring; therefore, individuals of better fitness are eventually obtained [5].

The TS method was introduced by Glover [6] as a high-level algorithm that uses other specific heuristics to guide the search. Its objective is to perform an intelligent exploration of the search space to prevent the tendency of getting trapped in local optima. The objective is thus to remedy one of the main issues of local search methods, namely a useless search in the neighbourhood of local optima—due to revisiting solutions or paths of solutions already explored—without further improvements. This is achieved by assigning the 'tabu' (forbidden) status to solutions visited in the recent search. In addition, TS is designed to be flexible so that the tabu status of solutions can be waived if they have been prohibited for an extended time or if they satisfy some aspiration criteria. The classification of some solutions as tabu is achieved through the intelligent use of adaptive memory, which is able to evolve and eventually change the status of tabu solutions [7].

In this study, given the locations of mesh routers and their traffic demands, we focus on the problem of determining which routers play the roles of gateways and the connectivity among them, i.e., the network topology, subject to the number of antennas allowed to be installed in a mesh router, the capacity of wireless links, and the maximum tolerable delay, such that the construction cost is minimal. This is referred to as the WMN design problem. The construction cost includes the cost of setting up gateways and also depends on the number of antennas used. The number of deployed gateways accounts for the major part of the construction cost because wiring may be not only difficult and disruptive but also expensive and time-consuming. Note that, in general, the greater the number of deployed gateways, the smaller is the number of required antennas in the network, i.e., it is necessary to achieve a trade-off between them. We aim to investigate the optimal network configuration, including the topology and the number of antennas and gateways required, such that the network construction cost is minimal. Note that, in order to further enhance survivability against node failure, the network topology requires that each mesh router have at least two node-disjoint paths to different gateways.

The remainder of this paper is organized as follows. In Section II, related works on the design of the WMN configuration are described. In Section III, the WMN network model is presented. The GA and its operators are considered in Section IV, and in Section V, the tabu search is described. Section VI outlines how the GA and TS are employed to solve the WMN design problem. Computational results of the proposed algorithms are provided in Section VII. Finally, Section VIII concludes the paper with pointers to future work.

II. RELATED WORK

Most proposals that deal with the design problem do not account for all parameters that have an impact on the design. Moreover, they assume the existence of a physical topology in which the characteristics of nodes (e.g., number of channels, number of radios, and range) are fixed. A simplified version of a generalized model for WMN design with an unfixed topology in which the characteristics and placement of nodes are not predefined is proposed in [8]. The aim of that work is to determine a WMN topology and configuration with a least cost that satisfies the requirements in terms of throughput and delay. For a fixed topology in which the locations of nodes and their characteristics are predefined, the WMN design problem is reduced to channel assignment and routing or gateway placements. The authors in [9, 10] and [11, 12] propose channel assignment algorithms to maximize throughput and satisfy end user demands while assuming a fixed topology. The authors in [13] and [14, 15, 16] (with support of quality-of-service requirements) propose techniques to place and minimize the number of gateways supporting a specific amount of traffic, while the characteristics of the network nodes are predefined (transmission power, number of radios, and number of channels).

Mesh topology is commonly considered in wide area networks (WANs) [17, 18]. The main WMN design problems relate to (1) placement of gateways; (2) determination of transmission power; and (3) channel assignment. If the placement/location of the nodes (e.g., routers/gateways) and their characteristics (i.e., number of radios per node, number of channels per radio, and transmission power) are fixed, the WMN design problem is reduced to channel assignment and routing. Because channel allocation in WMNs is an NP-hard problem [10], most design approaches propose mathematical models that are solved using linear programming and heuristics [9, 10, 19, 20, 21]. Accordingly, the location of the nodes is crucial in WMNs because it is directly related to efficiency and deployment costs. Additionally, the location problem is solved using linear programming and heuristics [14, 15, 16].

Existing approaches propose solutions that deal with only a part of the design problem (i.e., some parameters are omitted or, in the best case, are predefined/fixed). The authors in [22] calculate the per-node throughput for a predefined topology, including the location of gateways. The authors in [16] and [14, 15] propose techniques to place and minimize the number of gateways while supporting a specific amount of traffic to and from the Internet; the characteristics of nodes are fixed. The authors in [9, 10] and [12, 20] propose channel assignment algorithms to maximize throughput and satisfy end user demands while assuming a fixed topology. Unlike what has been proposed elsewhere in the literature, they exploit the relationship between design components for an unfixed topology; characteristics and placement of nodes are not predefined. The fact is that the design cost can be reduced but not minimized when other parameters are omitted.

Sen and Raman [16] introduce a variety of design considerations and a solution approach that breaks down the WMN planning problem into four 'more manageable' parts. These sub-problems are interdependent and solved by heuristics in a definite, significant order. Other related works [23, 24] deal with creating a WMN model, planning its parameters, and evaluating the solutions via linear programming. He etal. [24] propose mechanisms for optimizing the placement of integration points between the wireless and wired networks. They developed algorithms to provide optimal coverage by making informed placement decisions based on neighbourhood layouts, user demands, and wireless link characteristics. Amaldi etal. [2] propose other planning and optimization models based on linear programming. Their aim is to minimize network installation costs by providing full coverage for wireless mesh clients; accordingly, traffic routing, interference, rate adaption, and channel assignment are taken into account. Another cost-minimizing, topology planning approach is presented by So and Liang [23]. They propose an optimization framework that combines a heuristic with Benders decomposition to calculate the minimum deployment and maintenance cost of a given heterogeneous wireless mesh network. Furthermore, an analytical model is presented to investigate whether a particular relay station placement and channel assignment can satisfy the user demands and interference constraints.

Ghosh etal. [21] are the first to use GAs for wireless multi-hop optimization. They strive to minimize costs and maximize the link availability of a universal mobile telecommunications system (UMTS) network with optical wireless links to the radio network controllers. Along with Gosh etal., Badia et al. [25] use GAs for WMN joint routing and link scheduling. The packet delivery ratio is optimized with a dependency on frame length. They determine that GAs solve the given problems reasonably well and are also scalable, whereas exact optimization techniques are unable to find solutions for larger topologies. The performance of the GA is shown for a single-rate, single-channel, single-radio WMN. Vanhatupa etal. [26, 27] apply a GA for WMN channel assignment. Capacity, AP fairness, and coverage metrics are used with equal significance to optimize the network. The routing is fixed by using either shortest path routing or expected transmission times. In contrast to the works by Badia [25] and Vanhatupa [27], Vanhatupa etal. evaluate the performance of a multichannel, multi-radio, multi-rate WMN using both channel and route assignment.

In [28], path-oriented encoding is adopted. For each destination, a sink tree is constructed by connecting it to the source and all other destinations using the shortest (i.e., least-cost) paths. On the sink tree, each path from the tree root to a leaf node is referred to as a super path. With each iteration, the TS algorithm first generates a few neighbours of a multicast tree by separately replacing its one super path using a few randomly selected super paths. Then, among these new neighbours, the one with the best cost is selected; it is considered the new solution for the next iteration. If a super path is deleted in one iteration, then reintroducing the same super path to the current tree is deemed tabu. Assuming m is the number of destinations, there are m sink trees. Each candidate solution is just one combination of m paths from m sink trees. Therefore, the size of the candidate solution space is limited by all sink trees. The performance of the algorithm is hindered by the limited solution space to be explored.

III. NETWORK MODEL

Before explaining the research problem, we must first clarify some key concepts—antenna system, full-duplex emulation, traffic demand, and capacity links—that may help simplify and elucidate the problem.

An antenna system is one of the primary components for ensuring high performance of WMNs. Two types of antennas—omnidirectional and directional—can be used to construct wireless links. Omnidirectional antennas have been used as the underlying antenna technology in many WMNs and test beds. Omnidirectional antennas can be easily installed; however, severe interference limits the available bit rate. To reduce interference, nodes must be properly separated in space and frequency domain. To this end, multichannel mechanisms are typically used, which results in low spectrum efficiency. On the other hand, directional antennas are suggested for constructing WMNs. Directional antenna systems work well on the premise that transmitting and receiving antennas are accurately aimed at each other. Using directional antennas, interference can be significantly reduced; moreover, they provide an additional degree of freedom for allocating radio resources. Furthermore, directional antennas have a high antenna gain, which is helpful for increasing channel bit rate and reducing the error rate. Therefore, using directional antennas is relevant to WMNs.

In our proposed approach, directional antennas are used and interference is neglected. In fact, even when using directional antennas, if the mesh routers are closely located, interference may still occur. In this circumstance, interference can be eliminated using high-gain antennas or frequency diversity. In terms of system complexity, cost, and legal constraints, the number of antennas allowed to be installed in a mesh router is restricted. This limitation is a node degree constraint on topology [29].

Although nodes in a WMN can operate in either time division duplex (TDD) or frequency division duplex (FDD) mode, TDD is chosen here because it simplifies the frequency assignment problem, results in fewer required antennas, and has a higher spectrum efficiency in the face of asymmetric traffic, such as HTTP and FTP. Each mesh router has certain upstream and downstream traffic demands. Based on the flexibility of bandwidth allocation provided by TDD, downstream traffic demands can be integrated with upstream traffic demands so that the mesh router logically has only one traffic demand, which thereby simplifies the network model.

The traffic demand of a mesh router is composed of effective client traffic demands. For example, if a mesh router is used as a backhaul router of an IEEE 802.11b WLAN, then a traffic demand assignment of 6.2 Mbps is sufficient for the mesh router to serve the WLAN in any case. Gateways connect to the Internet via wireline links. It is reasonable to assume that the capacity of a wireline link is large; therefore, wireless links dominate the capacity and performance of a WMN. Because a gateway can directly serve its traffic demand by the wireline link, a gateway will show no traffic demand to the WMN. The capacity of a wireless link is confined by path loss and channel quality, and it suffers from attenuation and interference. Because channel quality varies with time, in designing a WMN, a link margin can be introduced as an estimation of channel quality degradation. We assume that link capacity can be fully utilized for data transmissions. Note that control overhead consumes some capacity, which reduces the effective capacity available for data transmission. Nevertheless, control overhead can be calculated and deducted in advance such that only the effective capacity is considered during network topology optimization.

Now, we can define the problem by way of a mesh network modelled as a graph, where mesh routers are represented by vertices and wireless links are represented by arcs. Notations adopted in the problem formulation are:

- R set of all mesh routers
- N number of mesh routers
- K maximum number of antennas allowed to be installed in a mesh router
- a_{uv} indicator function, which is 1 if a direct wireless link is formed between mesh routers *u* and *v*, and 0 otherwise: $a_{uu} = 0$
- λ_u traffic demand of mesh router *u*
- t_{uv} traffic load offered by mesh routers *u* to *v*: $t_{uv} \geq 0$, $t_{uu} = 0$
- c_{uv} link capacity of the wireless link between mesh routers *u* and *v*: $c_{uv} \geq 0$, $c_{uu} = 0$
- δ_u indicator function, which is 1 if mesh router *u* is a gateway and 0 otherwise
- σ_u cost on setting up mesh router *u* as a gateway
- D maximum tolerable delay
- d_u maximum delay of mesh router *u*

The WMN design problem is then formulated as:

$$Min \quad Z = \sum_{u=1}^{N}\left(\sum_{v=1}^{N} a_{vu} + \sigma_u \delta_u\right) \qquad (1)$$

such that

$$\forall u, v \in R$$

$$\sum_{v} a_{uv} \leq K \qquad (C1)$$

$$c_{uv} a_{uv} \geq t_{uv} \qquad (C2)$$

$$\left(\sum_{v} t_{uv} + \lambda_u\right) = (1 - \delta_u)\sum_{v} t_{vu} \qquad (C3)$$

$$\sum_{u}(1 - \delta_u)\lambda_u = \sum_{u}\sum_{v} \delta_u t_{vu} \qquad (C4)$$

$$a_{uv} - a_{vu} = 0 \qquad (C5)$$

$$d_u \leq D \qquad (C6)$$

The cost of setting up an antenna and setting mesh router *u* as a gateway is normalized as 1 and σ_u, respectively. Constraint (C1) represents the degree constraint on mesh routers, while constraint (C2) requires that the offered load of mesh routers *u* to *v* does not exceed the link capacity. Constraint (C3) requires traffic balance for each nongateway mesh router. Moreover, for a gateway mesh router, no traffic demand is present and a gateway does not offer traffic load to other mesh routers. Constraint (C4) requires that all traffic demands are served by gateways. Constraint (C5) means that a link is formed by two opposite antennas. Constraint (C6) requires that the maximum packet delay of each mesh router is within an acceptable range. Furthermore, for the sake of survivability, the WMN requires that each nongateway mesh router should have at least two node-disjoint paths to different gateways. Therefore, a WMN must meet the following survivability requirement:

$$\exists i, j \ni (1 - \delta_u)\, n_{i,j}^{u} = 0 \qquad (C7)$$

where $n_{i,j}^{u}$ denotes the number of common nodes among mesh router *u*'s *i*th and *j*th paths.

IV. GENETIC ALGORITHMS

A GA is a metaheuristic technique that is used to solve optimization problems by imitating natural selection; i.e., the process of adaptation to the environment performed by living beings [30]. GAs are an appealing approach to solving the complex problem outlined in the previous section. A GA determines not a single solution but a whole 'population' of 'individuals,' which are candidate solutions to the problem. The distinctive features of each individual are coded into a structure called a 'chromosome'. The chromosome is a string of genes, whose values can be chosen from within a set of symbols. An application-dependant process generates the individual by decoding its chromosome. The symbols used as values of the genes are typically binary, integer, or real numbers, depending on the nature of the problem. Once an individual is generated, a fitness function is used to evaluate its fitness as a solution to the problem. Low values of fitness function are typically assigned to the most fit individuals (minimization problem).

A GA starts with an initial population generated either randomly or with some heuristic approach that exploits the knowledge of an expert in the problem domain. The algorithm then proceeds in steps called generations. At each generation t, a new population *P(t + 1)* is evolved from *P(t)*. As generations pass, the population should globally improve on account of the application of genetic operators that mimic natural evolutionary mechanisms. To this end, the most fit individuals are chosen from *P(t)* (selection) to be mated (crossover) and slightly modified (mutation) so as to create the new population *P(t+1)*. The selection operator is used to decide which individuals in *P(t)* should be chosen to generate *P(t+1)*. Optionally, an elite from among the selected individuals (i.e., a small number of the best performing individuals) survives and is moved from *P(t)* to *P(t+1)* with no change. The crossover operator consists in choosing some of the individuals and mating them. In other words, it substitutes them with their children; i.e., individuals generated by mixing the genetic material in the parents' chromosomes. The actual implementation of a crossover operation greatly depends on the coding schema of the chromosome. Finally, the mutation operator introduces some new genetic material in the population by randomly modifying the values of some genes. Again, different kinds of mutation operations can be defined to handle different sets of symbols. The population continues to evolve until a stopping criterion is fulfilled, with the simplest being a maximum number of generations. In addition, GAs are interesting for practical purposes because they can be regarded as rapid procedures for identifying a 'good enough' solution to the problem [31]. This gives them an advantage with respect to exact techniques because any solution produced by a GA is directly applicable.

Therefore, a GA can be used to operate in WMN design in which the solution may be iteratively updated.

V. TABU SEARCH

TS [6, 32] is a metaheuristic method for combinatorial optimization that was proposed by Fred Glover in 1986. It is a dynamic neighbourhood search technique that employs memory to drive the search by escaping from local optima and preventing cycling. The principal characteristic of TS lies in its systematic use of flexible and adaptive memory, which tracks information of the previously visited solutions. This technique contrasts with most commonly used search techniques, which retain only the value of the objective function that corresponds with the best solution visited to that point. The TS algorithm begins by randomly generating the first solution called *Sol*; it then creates a tabu list, which is a short-term set of the solutions that have been visited in the recent past. From that point, it records *Sol* in the tabu list. Next, it generates neighbourhood solutions near *Sol* and selects the best solution *Sol** according to the objective function of the problem. Upon seeking, if the *Sol** does not exist in the tabu list, then *Sol** is added to the tabu list and *Sol** is set as the current solution *Sol*; otherwise, it generates another neighbourhood solution near *Sol*. This process continues until the termination condition reaches a specified number of iterations. The main, formal elements of the algorithm are as follows:

- Solution representation: Each feasible solution to the optimization problem must have a unique representation within the search space.
- Cost function: A function of cost that maps each feasible solution into a value representing its optimization cost. The goal of the algorithm is to find a solution that minimizes this value.
- Neighbourhood: A function that maps each feasible solution into a set of other solutions. Each time the algorithm must consider a new solution, it is chosen from the neighbourhood of the current solution.
- Tabu list: A list containing the last moves carried out, which, for this reason, are forbidden. In general, a solution obtained from the current solution with a move contained in the tabu list cannot be a member of the neighbourhood of the current solution.
- Termination criterion: The algorithm stops when the termination criterion is satisfied [33]. Therefore, TS can be used in the WMN design problem.

VI. WMN TOPOLOGY DESIGN PROBLEM: TWO APPROACHES

We can now begin to illustrate how the GA and TS solve the WMN design problem. We must clarify the inputs and definitions used in our two proposed algorithms. Given the locations of mesh routers and their traffic demands, the network configuration can be designed; the two algorithms are proposed to address the WMN design problem. A network configuration is called a feasible network configuration (FNC) if, and only if, all constraints—i.e., from (C1) to (C7)—are met. The two algorithms determine the FNCs by routing paths for traffic demands. Then, they check predefined gateway sets to determine whether an FNC can be found.

The two algorithms serve to discover a set of gateways and the sequence to route paths for traffic demands such that the optimal solution (least cost) is obtained. Let g denote the number of gateways used. Because it is not known in advance how many gateways are required, the two algorithms search for an FNC using one gateway—i.e., $g = 1$—at the beginning; g is increased by one each time no FNC is obtained using g gateways. These processes continue until an FNC is obtained [29]. Given g, the two algorithms arbitrarily select g gateways and then route the paths for traffic demands. If all traffic demands are satisfied, the two algorithms proceed to verify if the network configuration meets the survivability requirement (C7); i.e., each mesh router must have at least two node-disjoint paths to different gateways. If true, then an FNC is obtained. Otherwise, the two algorithms add a gateway and repeat the process until the FNC is obtained.

The order of traffic demands for routing paths is specified in the routing sequence. Dijkstra's algorithm [29] is employed to find the path from the source mesh router to either of the gateways. If the capacity of the path can support the traffic demand, the traffic demand is satisfied by the path, and the link capacity along the path is deducted accordingly. Otherwise, the path can satisfy the traffic demand even if it transfers the packet in parts and the maximum tolerable delay is accepted; Dijkstra's algorithm will again be triggered if the delay is not accepted.

At the start of the two algorithms, all links are potential; that is, they are not determined. Links are determined only when required according to the path reported by Dijkstra's algorithm. If a path passes through a potential link, then the potential link becomes determined. Once a potential link is determined, the number of antennas used by the two end mesh routers of the given link is increased by one. In this way, we can eventually obtain a network configuration that satisfies all traffic demands. The paths for traffic demands are determined only in the design phase of the WMN. Mesh routers may select the optimal paths for their traffic demands with respect to the specific metric, quality-of-service requirement, and current traffic load at runtime.

A. Genetic algorithm approach

The applicability of GAs to the resolution of many computational combinatorial optimization problems has been shown. Needless to say, GAs are strong candidates for efficiently solving the WMN design problem.

To solve various, real-world problems, there are many factors to be considered when employing a GA, such as encoding methods, initial populations, selection, the selection of fitness function, crossover operation, mutation operation, and choice of parameters. These operators are detailed below.

1) Encoding

Encoding is the most fundamental operation in a GA. Each mesh router is labelled a unique number ranging from 1 to *N*, where *N* is the number of mesh routers. The chromosome, as shown in Fig. 1, comprises two parts. The first is the gateway indication component, which uses a binary string to indicate whether the corresponding mesh router is a gateway. The second part of the chromosome is the sequence component, which uses integers ranging from 1 to *N* to indicate the sequence for routing the path of the corresponding traffic demand. For example, a chromosome of 11000 13245 for a five-node network implies that mesh routers 1 and 2 are gateways, and it routes paths for the traffic demands of mesh routers 3, 4 and 5. Chromosomes are initialized by arbitrarily selecting gateways and sequence numbers.

1	1	0	0	0	1	3	2	4	5
First part					Second part				

Figure 1. Chromosome encoding

2) Fitness function

The chromosome that yields an FNC is deemed a successful chromosome; otherwise, it is a failed one. In the algorithm, the fitness value of a chromosome is evaluated by *fitness = 1/Z*, where *Z* is the objective function defined in (1). Therefore, a chromosome that represents a lower-cost network configuration will have a larger fitness value.

3) Selection

A pair of chromosomes is selected by using ranking selection, which sorts the chromosomes according to fitness value and then ranks them. Every chromosome is allocated a selection probability with respect to its rank. Rank selection is an explorative selection technique, which prevents a too-rapid convergence.

4) Crossover

The simplest approach is single-point crossover, whereby paired individuals are each cut at a randomly chosen crossover site; the portions after the cuts are exchanged to form two new children individuals. Fig. 2 illustrates the single-point crossover implementation.

	First part					Second part				
parent 1	1	1	0	0	0	2	1	4	3	5
parent2	0	1	1	0	0	1	3	2	5	4
rearrange parent 1	1	1	0	0	0	3	1	4	2	5
child1	1	1	0	0	0	3	1	2	5	4
child2	0	1	1	0	0	1	3	4	2	5

Figure 2. Single-point crossover

Parent chromosomes are rearranged to protect new children from gene (router or gateway) duplication.

5) Mutation

To mutate, the chromosome exchanges two randomly selected genes in the gateway indication and sequence parts, respectively, as shown in Fig. 3.

First part					Second part				
0	1	1	0	0	1	3	4	2	5
0	1	1	0	0	1	5	4	2	3

Figure 3. Mutation process

The choice of crossover probability P_c, as well as mutation probability P_m, is critical to the GA. In Section VII, we will search the best values for them in different networks.

6) Overall algorithm

The steps of the overall algorithm for the GA to solve the WMN design problem are as follows:

Step 1: Set the parameters. Set the population size (Pop_size), (P_c), (P_m) and the maximum iteration (maxit), and initialize gateway number $g = 1$ and iteration number $I = 1$.

Step 2: Increase the number of gateways by one: $g = g + 1$.

Step 3: Initialization:

- Randomly generate the initial population that has (Pop_size) chromosomes.
- Route path for traffic demands for each chromosome.
- Calculate the fitness for all chromosomes.
- Save the best chromosome of the current iteration.

Step 4: Check for the best chromosome; if it does not have a successful chromosome go to Step 2.

Step 5: Select candidate networks from the current population by the rank-selection method.

Step 6: Perform the crossover and mutation to obtain children candidate networks according to P_c and P_m, respectively.

Step 7: Establish the new population. Replace the parents with children.

Step 8: Route the path for traffic demands and calculate fitness for each chromosome in the new population.

Step 9: Obtain the best chromosome of the new population and compare the current best chromosome with the best one of the previous iteration. If it is better, it replaces the best chromosome of the previous iteration.

Step 10: Perform the terminating test. If $I <$ maxit, set $I = I + 1$, and go to Step 4 for the next iteration; otherwise, terminate.

The required optimal network (least cost and number of gateways *g*) will be the one represented by the best chromosome of all iterations.

B. Tabu search approach

Among the possible heuristics that can solve our optimization problem, we choose the TS method. This method is a local search optimization technique that strives to minimize cost function *F(S)*—where *S* represents a parameter vector—by iteratively moving from solution *S* to solution *S** in the neighbourhood of S until a stopping criterion is satisfied (until a predetermined number of iterations is reached). A neighbourhood is constructed to identify adjacent solutions that can be reached from the current solution. A simple swap is defined, which swaps elements in solution S. The steps of the overall algorithm for TS to solve the WMN design problem are as follows:

Step 1: Set the parameters. Set the tabu list (TL) and the maximum iteration (maxit) and initialize gateway number $g = 1$ and iteration number $I = 1$.
Step 2: Increase the number of gateways by one: $g = g + 1$.
Step 3: Generate a random solution (*Sol*) that represents a network similar to the chromosome structure in the GA method.

0	1	0	1	0	1	3	2	4	5

First part — Second part

Figure 4. Solution in TS

Step 4: Route the path for traffic demands and calculate fitness for current solution *Sol.*
Step 5: Add *Sol* to the tabu list (TL).
Step 6: Generate neighbourhood solutions near *Sol*. We can use the mutation method in the GA to produce neighbourhood solutions. For example, one of the neighbourhood solutions can be represented as shown in Fig. 5.

0	1	0	1	0	3	1	2	4	5

First part — Second part

Figure 5. Neighborhood solution

Step 7: Route the path for traffic demands and calculate fitness for all neighbourhood solutions.
Step 8: Determine the best solution *Sol** from the neighbourhood solutions.
Step 9: Check the best solution Sol^*; if it does not have a successful solution, go to Step 2.
Step 10: Obtain the best solution of the new iteration and compare the current best solution with the best one of the previous iteration. If it is better, it replaces the best solution of the previous iteration.
Step 11: Verify if Sol^* is not found in the tabu list. Add Sol^* in the tabu list and set Sol^* as current solution *Sol;* otherwise, go to Step 6.
Step 12: Perform the terminating test. If $I <$ maxit, set $I = I + 1$, and proceed to Step 4 for the next iteration; otherwise, terminate.

The required optimal network (least cost and number of gateways g) will be the one represented by the best solution of all iterations.

VII. COMPUTATIONAL RESULTS

We implemented our application in Visual C++ 2010. The algorithm was executed on a PC with an Intel core i5 2.40 GHz processor and 4 GB of RAM. The experimental results are shown in Table I. We evaluated the performance of the proposed algorithms by using three test cases that represent different network configurations. A poor connection meant that the network had a low link capacity. First, we provide values for GA and TS parameters to serve as a comparison study in this section. The GA parameters are Pop_size = 20, $P_c = 0.4$, and $P_m = 0.4$. The TS parameter is the tabu list (TL = 5). We executed two algorithms at a different number of iterations for each test case.

TABLE I. EXPERIMENTAL RESULTS

Network	Iteration number	Comparison	GA	TS
20 routers–normal connection	10	Gateway	2	2
		Cost(units)	550	554
		Time(ms)	1861	77749
	50	Gateway	2	2
		Cost(units)	548	554
		Time(ms)	8799	454982
	100	Gateway	2	2
		Cost(units)	536	542
		Time(ms)	18396	826196
20 routers–poor connection	10	Gateway	6	6
		Cost(units)	1697	1351
		Time(ms)	466	83000
	50	Gateway	6	6
		Cost(units)	1595	1351
		Time(ms)	2173	371743
	100	Gateway	6	6
		Cost(units)	1436	1351
		Time(ms)	3999	2.4e+006
50 routers–normal connection	10	Gateway	2	2
		Cost(units)	828	830
		Time(ms)	28945	9.0+006
	50	Gateway	2	2
		Cost(units)	809	812
		Time(ms)	130791	1.1e+008
	100	Gateway	2	2
		Cost(units)	808	808
		Time(ms)	261935	3.4e+008

Important observations can be made from the results shown in Table I:

- An increase in the number of iterations leads to an improvement (minimizing of) network costs in GA and TS.
- In a normal connection network (link capacity is high), the GA is better than TS. However, in a poor network connection, TS is better than the GA and the optimal value may not be changed.
- TS employs a long running time, which is difficult to use in a large-size network and it increases the number of iterations. It can be continued for several hours.
- A poor-connection network requires a greater number of gateways to overcome weakness in link capacity.
- In a large-size network with a normal connection, a significant approach in optimization cost is found between the GA and TS.

Because the GA and TS are based on randomization in their operations, we can find an insignificant number of solutions that are not compatible with previous findings. We recommend using the GA with a small- or large-size network with a high link capacity. TS can be used with small networks and poor connections. If running time is desired as an effect factor in solving the optimization problem, the GA is the only choice.

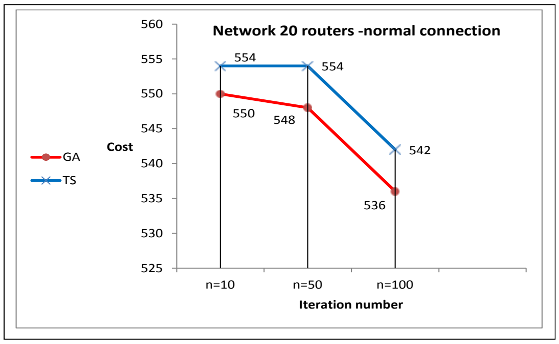

Figure 6. Network of 20 routers – normal connection cost

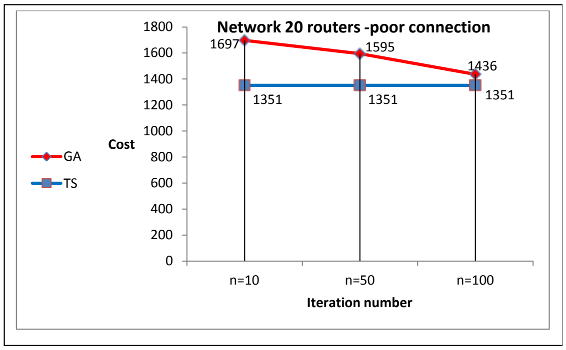

Figure 7. Network of 20 routers – poor connection cost

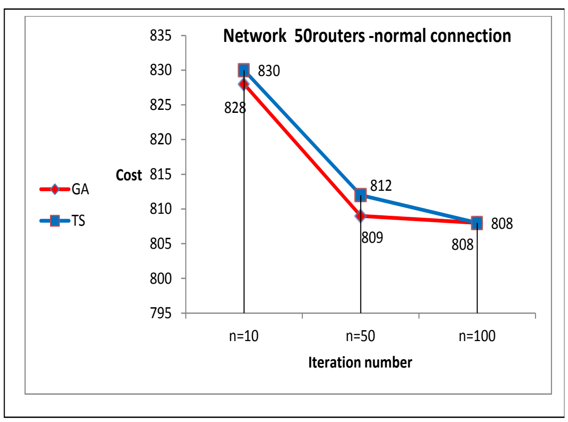

Figure 8. Network of 50 routers – normal connection cost

A. *Optimization with GA parameters*

Parameters in the GA have a signification impact on solving the optimization problem. Therefore, we strived to find the best value for each parameter in the WMN design problem under one of the previous test cases.

1) *Probability of crossover*

Probability of crossover specifies the rate of crossover (mating) occurring between two chromosomes. We changed the values of Pc to search for the best value for the optimization problem (to minimize cost); we found the best value when P_c = 0.2, as shown in Table II. We entered values for other parameters (iteration number = 100, Pop_size = 20, and P_m = 0.4) with a 20-router network and a normal connection.

TABLE II. CROSSOVER PROBABILITY OPTIMIZATION

P_c	Cost (units)	P_c	Cost (units)
0.1	544	0.6	546
0.2	538	0.7	542
0.3	542	0.8	542
0.4	540	0.9	552
0.5	544		

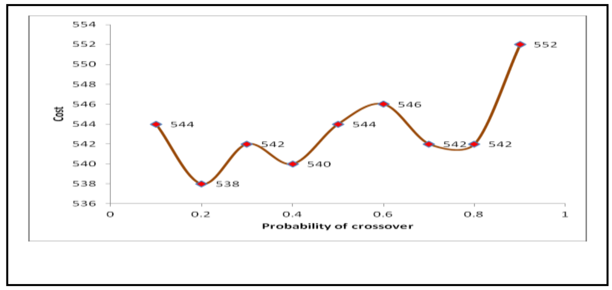

Figure 9. Cost – probability of crossover relation

2) *Probability of mutation*

Probability of mutation specifies how often parts of chromosomes will be mutated. We changed the values of P_m to search for the best value for the optimization problem (to minimize cost); we found the best value when P_m = 0.3, as shown in Table III. We entered values for other parameters (iteration number = 100, Pop_size = 20 and P_c = 0.2) with a 20 router-network and a normal connection.

TABLE III. MUTATION PROBABILITY OPTIMIZATION

P_m	Cost (units)	P_m	Cost (units)
0.1	542	0.6	542
0.2	548	0.7	546
0.3	536	0.8	540
0.4	538	0.9	538
0.5	538		

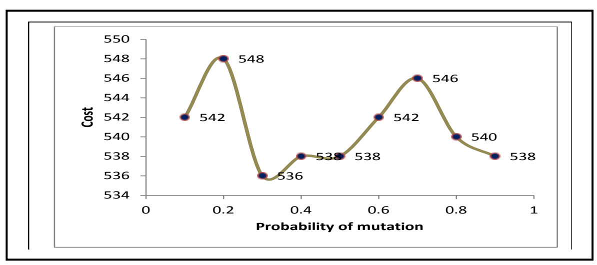

Figure 10. Cost – probability of mutation

3) *Population size*

Population size determines how many chromosomes are available and, therefore, how much genetic material is available for use during the search. We changed the values of population size of the generation to search for the best value for the optimization problem (to minimize cost); we found the best value when Pop_size = 90, as shown in Table IV. We entered values for other parameters (iteration number = 100, P_m = 0.2, P_c = 0.3) with a 20-router network and a normal connection.

TABLE IV. POPULATION SIZE OPTIMIZATION

Population size	Cost (units)	Population size	Cost (units)
10	552	60	538
20	540	70	540
30	538	80	534
40	542	90	532
50	544	100	536

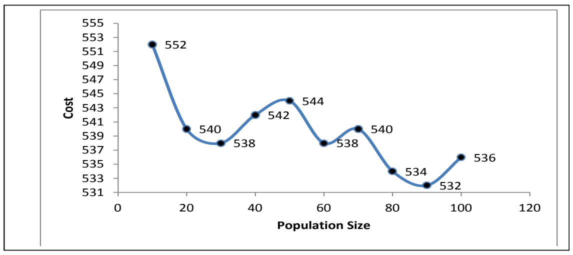

Figure 11. Cost – population size relation

B. Optimization with TS parameters

In TS, we performed experiments to examine the relationship between TL and the optimization problem (to minimize cost). As shown by information in Table V, the best cost was at TL = 8 with a 20-router network and a normal connection when the number of iterations was 100.

TABLE V. TABU LIST OPTIMIZATION

TL	Cost (units)	TL	Cost (units)
2	542	7	544
3	540	8	538
4	540	9	540
5	540	10	544
6	540		

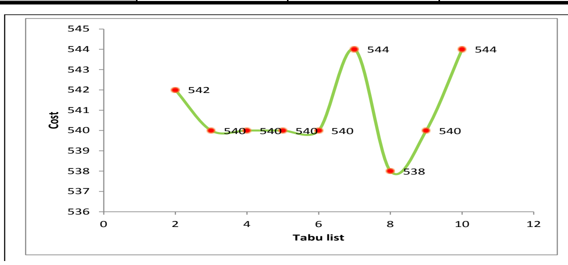

Figure 12. Cost – tabu list

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a GA and TS for the WMN design problem. Our functional objective was to minimize cost and search gateways of the WMN under constraints. Performance comparisons of the GA and TS under different networks were presented. In addition, the running time required for these algorithms was evaluated. We concluded that the GA and TS were able to minimize cost. The GA was better than TS in a network in which link capacity was high; however, TS was better in a network in which link capacity was low. Further, we examined the optimal value for parameters in the GA and TS. The global optimal could not be determined because the functional objective (to minimum cost) exhibited an irregular manner when changing parameter values. In future work, we intend to propose another metaheuristic method to further advance the WMN design problem.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang, Wireless mesh networks: a survey, Computer Networks 47(4), pp. 445–487, 2005.

[2] E. Amaldi, A. Capone, M. Cesana, I. Filippini, and F. Malucelli, Optimization models and methods for planning wireless mesh networks, Computer Networks, vol. 52, no. 11, pp. 2159–2171, August 2008.

[3] Ch. Chen and Ch. Chekuri, Urban Wireless mesh network planning: the case of directional antennas, Tech Report No. UIUCDCS-R-2007-2874, Department of Computer Science, University of Illinois at Urbana-Champaign, 2007.

[4] J. Holland, Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, 1975.

[5] Z. Michalewicz, Genetic algorithms + data structures = evolution program, 3rd Edition, Springer-Verlag, Berlin, 1996.

[6] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers and Op. Res., vol. 5, pp. 533–549, 1986.

[7] X. Fatos, S. Christian, B. Admir, T. Makoto, A tabu search algorithm for efficient node placement in wireless mesh networks, 2011 Third International Conference on Intelligent Networking and Collaborative Systems, pp. 53–59, 2011.

[8] A. Beljadid, A. S. Hafid, and M. Gendreau, Optimal design of broadband wireless mesh networks, in Proc. of the 50th International Conference on GLOBECOM 07, 2007.

[9] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy, Optimization models for fixed channel assignment in wireless mesh networks with multiple radios, IEEE SECON 2005, pp. 463–474, 2005.

[10] A. Raniwala K. Gopalan and T-C. Chiueh, Centralized channel assignment and routing algorithms for multi channel wireless mesh networks, Comput. Commun. Rev., vol. 8, no. 2, pp. 50–65, 2004.

[11] M. Alicherry, R. Bhatia, and L. E. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, MobiCom '05, pp. 58–72, 2005.

[12] P. Kyasanur and N. Vaidya, Routing and interface assignment in multichannel multi-interface wireless networks, WCNC, 2005.

[13] B. Aoun, R. Boutaba, Y. Iraqi, G. Kenward, Gateway placement optimization in wireless mesh networks with QoS constraints, IEEE JSAC, vol. 24, Issue 11, pp. 2127–2136, Nov. 2006.

[14] J. L. Qiu, R. Chandra, K. Jain, and M. Mahdian, Optimizing the placement of integration points in multi-hop wireless networks, Proc. ICNP'04, pp. 271–282, Oct. 2004.

[15] Nomura, N. Funabiki, and T. Nakanishi, A proposal of access-point channel assignment algorithm and an evaluation of access-point allocation algorithm in wireless infrastructure mesh networks, in Proc. 3rd ANWS, pp. 113–116, Jan. 2006.

[16] S. Tajima, T. Higashino, N. Funabiki, and S. Yoshida, An internet gateway access-point selection problem for wireless infrastructure mesh networks, in Proc. of the 7th International Conference on MDM '06, 2006.

[17] A. Kashyap, K. Lee, and M. Shayman, Rollout algorithms for integrated topology control and routing in wireless optical backbone networks, ISR Tech. Report TR2003-51, 2003.

[18] Y. Ding and L. Xiao, Channel allocation in multi-channel wireless mesh networks, Computer Communications, vol. 34, pp. 803–815, 2011.

[19] L. Hui, A novel QoS routing algorithm in wireless mesh networks, TELKOMNIKA, vol. 11, no. 3, pp. 1652–1664, ISSN: 2087-278X, March 2013.

[20] M. Kodialam and T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, MobiCom '05, pp. 73–87, 2005.

[21] S. Ghosh, P. Ghosh, K. Basu, and S. K. Das, GaMa, An evolutionary algorithmic approach for the design of mesh-based radio access networks, in LCN ’05: Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary, Washington, DC, USA, pp. 374–381, November 2005.

[22] J. Jun and M. L. Sichitiu, The nominal capacity of wireless mesh networks, IEEE Wireless Communications, Oct 2003.

[23] A. So and B. Liang, Minimum cost configuration of relay and channel infrastructure in heterogeneous wireless mesh networks, in Networking, Atlanta, GA, USA, pp. 275–286, May 2007.

[24] B. He, B. Xie, and D. P. Agrawal, Optimizing deployment of internet gateway in wireless mesh networks, Computer Communications, vol. 31, no. 7, pp. 1259–1275, 2008.

[25] L. Badia, A. Botta, and L. Lenzini, A genetic approach to joint routing and link scheduling for wireless mesh networks, Elsevier Ad Hoc Networks Journal, vol. Special issue on Bio- Inspired Computing, p. 11, April 2008.

[26] T. Vanhatupa, M. H¨annik¨ainen, and T. D. H¨am¨al¨ainen, Genetic algorithm to optimize node placement and configuration for WLAN planning, 4th International Symposium on Wireless Communication Systems, 2007. ISWCS 2007, Trondheim, Norway, pp. 612–616, October 2007.

[27] T. Vanhatupa, M. H¨annik¨ainen, and T. D. H¨am¨al¨ainen, Performance model for IEEE 802.11s wireless mesh network deployment design, Journal of Parallel and Distributed Computing, vol. 68, no. 3, pp. 291–305, March 2008.

[28] Y. Habib, A. Abdulaziz, M. Sadiq, A. Muhammad, QoS-driven multicast tree generation using tabu search, Comput. Commun, vol. 25, pp. 1140–1149, 2002.

[29] H. Chun-Yen, C. Jean-Lien, W. Shun-Te, H. Chi-Yao, Survivable and delay-guaranteed backbone wireless mesh network design, J. Parallel Distrib. Comput. 68, pp. 306–320, 2008.

[30] K. A. De Jong, Evolutionary computation: A unified approach, MIT Press, 2006.

[31] L. Ting-Yu, H. Kai-Chiuan, H. Hsin-Chun, Applying genetic algorithms for multiradio wireless mesh network planning, IEEE Transactions on Vehicular Technology, vol. 61, no. 5, pp. 2256–2270, June 2012.

[32] S. Martins , C. Ribeiro, Metaheuristics and applications to optimization problems in telecommunications, Handbook of Optimization in Telecommunications, M. Resende and E.P. Pardalos, Eds. New York: Springer Science + Business Media, pp. 103–128, 2006.

[33] S. Kazem, K. Maryam, Y. Mohammed, D. Houssien, Localization in wireless sensor networks using tabu search and simulated annealing, Computer and Automation Engineering (ICCAE), 2nd International Conference On Singapore, vol. 2, pp. 752–757, 2010.