Solving the Wireless Mesh Network Design Problem using Genetic Algorithm and Simulated Annealing 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
Mesh clients, mesh routers and gateways are components of Wireless Mesh Network (WMN). In WMN, gateways connect to Internet using wireline links and supply Internet access services for users. Multiple gateways are needed, which take time and cost budget to set up, due to the limited wireless channel bit rate. WMN is a highly developed technology that offers to end users a wireless broadband access. It offers a high degree of flexibility contrasted to conventional networks; however, this attribute comes at the expense of a more complex construction. Therefore, a challenge is the planning and optimization of WMNs. This paper concentrates on the challenge using a genetic algorithm and simulated annealing. The genetic algorithm and simulated annealing enable searching for a low-cost WMN configuration with constraints and determine the number of used gateways. Experimental results proved that the performance of the genetic algorithm and simulated annealing in minimizing WMN network costs while satisfying quality of service. The proposed models are presented to significantly outperform the existing solutions.

Keywords
Wireless mesh networks, genetic algorithms, simulated annealing, topology design.

1. INTRODUCTION
Wireless Mesh Networks (WMNs) [1] have turned into significant networking infrastructure because of their high-speed for wireless Internet connectivity and low cost. There are two kinds of nodes for WMN: mesh routers and mesh clients. Mesh routers are analogous to traditional routers but integrate extra services to support mesh networking. They are well organized with various interfaces to accommodate different wireless technologies. Another characteristic that discriminates mesh routers from conventional routers is their capability to provide the same coverage with much less transmitter power through multi-hop communications. In addition, mesh routers can be established on general-purpose or dedicated machines. On the other side, mesh clients are equipped with tasks essential for mesh networking and can also serve as routers; however, they are incapable of functioning as bridges or gateways. The wireless interface of mesh client with hardware and software platforms is simpler than the mesh router interface. WMNs depend on a mesh topology in which every node (representing a server) is connected to one or more nodes, thereby allowing information transmission in extra one path. Mesh topology does not require a central node comparing to other topologies. These attribute permits mesh networks to be self-healing. Consequently, these types of networks are powerful to possible server node failures and more reliable.

On account of the above characteristics, rapid development of WMNs has been further compelled by their low associated costs, such as avoiding the expense of deploying and maintaining wired Internet infrastructures. This makes WMNs an economical option for supporting wireless Internet connectivity, especially in developing countries. WMNs applications include those for metropolitan area networks; urban areas; local wireless mesh networks; corporate and enterprise networks; neighbourhood, community, and home networks; surveillance, transport, and medical systems; building automation; among others [2]. Many optimization problems have demonstrated their applicability to the effective design of WMNs. These problems relate to optimize user coverage, network connectivity, and stability among other aspects. Their resolution is vital for optimizing network performance [3]. This paper proposes and evaluates genetic algorithms (GAs) [4,5] and simulated annealing (SA) [6] for near-optimally solving one of these problems—minimizing costs—as it interacts to WMN design.

GAs are evolutionary algorithms designed to apply the selection operation as it happens in nature. GAs begin from an initial population of 'individuals'—i.e., suitable solutions of a problem—with each solution having correlating fitness value that mentions how fit it is compared to the others. Thus, as in nature, where there are natural processes of selection, reproduction, and mutation, a GA practises a similar manner of evaluation, selection, crossover, mutation, and replacement, thereby engendering the next generation of individuals. The procedure is repeated until it reached a number of generations or best solution not changed along generations. The best characteristics of parents are passed to offspring; therefore, chromosomes of better fitness are finally obtained [5].SA technique is a probabilistic search method widely used in various areas of studies such as mathematics, computer science, engineering and operation research. In recent years SA has been applied to solve global optimization problems. This process is based on analogy from thermodynamics where a system is slowly cooled in order to achieve its lowest energy state. SA has the ability of escaping local minima by incorporating a probability function in accepting or rejecting new solutions. The main advantage of SA method is that it does not need large computer memory [6, 7].

In this study, the locations of mesh routers and their traffic demands are given, the paper focuses on the problem of determining which routers act the functions of gateways and the connectivity among them, i.e., the network topology, subject to the number of antennas accepted to be established in a mesh router, the maximum tolerable delay, and the capacity of wireless links such that the construction cost is minimal. This is mentioned as the WMN design problem. Here, the network cost consists of the cost of setting up gateways and also depends on the number of antennas used. The major part of network cost is the number of deployed gateways because wiring may not only be 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 essential to achieve a trade-off between them. The aim is to explore the optimal network configuration, including the topology, the number of antennas and gateways required, such that the network construction cost is minimal. The network topology ensures for each mesh router has at least two node-disjoint paths to different gateways for support survivability against node failure.

The remainder of this paper is arranged as follows. In Section 2, related works on the design of the WMN configuration are described. In Section 3, the WMN network model is presented. The GA and its operators are considered in Section 4, and in Section 5, the simulated annealing is described. Section 6 outlines how the GA and SA are used to solve the WMN design problem. Computational results of the proposed algorithms are described in Section 7. Finally, Section 8 concludes the paper with indicators to future work.

2. RELATED WORK

Most of the proposals that handle the network design problem do not have regard for all parameters that have an influence on the design. Moreover, they suppose the presence of a physical topology in which the attributes of nodes (e.g., number of radios, range, and number of channels) are fixed. A basic version of a global form for WMN design with an unfixed topology in which the attributes and placement of nodes are not predefined is proposed in [8]. The aim of that search is to assign a WMN configuration and topology with a least cost that satisfies the demands in terms of delay and throughput.

The WMN design problem is reduced to channel assignment and routing or gateway placements for a fixed topology in which the locations of nodes and their characteristics are predefined. The authors in [9, 10] and [11, 12] suggests channel assignment algorithms to satisfy end user demands and maximize throughput while assuming a fixed topology. The authors in [13] and [14, 15, 16] (with support of quality-of-service requirements) suggest methods to minimize and place the number of gateways supporting a certain amount of traffic, while the features of the network nodes are predefined (number of channels, transmission power, and number of radios).

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

Many approaches suggest solutions that deal with only a branch of the design problem (i.e., some parameters are neglected or, in the best case, are predefined/fixed). The authors in [22] compute the per-node throughput, including the location of gateways for a predefined topology. The authors in [16] and [14, 15] suggest methods to place and minimize the number of gateways while supporting a specific amount of traffic to and from the Internet; the attributes of nodes are fixed. The authors in [9, 10] and [12, 20] suggest channel assignment algorithms while assuming a fixed topology to satisfy end user demands and maximize throughput. They use the relationship between design elements for an unfixed topology; attributes and placement of nodes are not predefined, unlike what has been proposed elsewhere in their work. The fact is that when other parameters are neglected, the design cost can be reduced but not minimized.

Sen and Raman [16] present a diversity of design regards and a solution method that breaks down the WMN planning problem into four 'more manageable' components. These sub-problems are mutually dependent and solved by heuristics in a significant, definite order. Other related works [23, 24] handle with generating a WMN model, organizing its parameters, and performing the solutions by linear programming. He et al. [24] suggest methods for optimizing the placement of combination points between the wired and wireless networks. They improved algorithms to supply optimal coverage by causing informed placement decisions on wireless link characteristics, user demands and neighbourhood layouts. Amaldi et al. [2] suggest other designing and optimization samples based on linear programming. Their objective is to minimize network installation costs by improving full coverage for wireless mesh clients; accordingly, rate adaption interference, traffic routing, and channel assignment are taken into account. So and Liang [23] have presented another cost-minimizing topology planning method. They suggest an optimization framework that merges a heuristic with Benders decomposition to compute the minimum maintenance and deployment cost of a given diverse wireless mesh network. Furthermore, an analytical model is showed to investigate whether a channel assignment and particular relay station placement can satisfy the interference constraints and user demands.

Ghosh et al. [21] apply GAs for solving wireless multi-hop optimization problem. They strive to maximize the link availability and minimize costs of a universal mobile telecommunications system (UMTS) network with optical wireless links to the radio network controllers. Along with Gosh et al., Badia et al. [25] apply GAs for WMN link scheduling and joint routing. They determine that GAs solve the given problems reasonably well and are also scalable, whereas optimization methods are incapable of obtaining solutions for wide topology networks. The performance of the GA is described for a single-channel, single-radio, single-rate WMN. Vanhatupa et al. [26, 27] suggest a GA for WMN channel assignment. Capacity, AP fairness, and coverage metrics are operated with equal impact to optimize the network. The routing is fixed, by using either expected transmission times or shortest path routing. In contrast to the works by Badia [25] and Vanhatupa [27]. Vanhatupa et al. evaluated the performance of a multi-rate, multi-radio, multichannel WMN operating both route and channel assignment.

In [28], they suggested the routing and channel assignment for dynamic traffic in WMNs. They assumed the static channel assignment strategy to the network interfaces. The problem is organized into two sequential stages. The first is to determine channels to interfaces while the second is to assign the route for each coming traffic demand. They suggested a Mixed Integer Linear Programming (MILP) formulation to the problem and improve a SA establish channel assignment algorithm for the channel assignment. The shortest path routing is assumed for the dynamic traffic. In other works, they suggested and evaluated a SA method to placement of mesh router nodes in WMNs. The optimization problem uses two maximization objectives, namely, user coverage and the size of the giant component in the network. Two objectives were critical to deployment of WMNs. They have experimentally evaluated the SA algorithm through a benchmark of produced instances, altering from small to large size, and capturing different features of WMNs such as topological placements of mesh clients. Their experimental results illustrated the performance of the SA method for the placement of mesh router nodes in WMNs [29].

3. NETWORK MODEL

Before explaining the research problem, some key concepts should be clarified—antenna system, full-duplex emulation, traffic demand, and capacity links—that may help simplify and elucidate the problem.

One of the main components for assuring high performance of WMNs is an antenna system. To construct wireless links can be used one of the two types of antennas—omnidirectional and directional. Omnidirectional antennas have been employed as the essential antenna technology in different WMNs and test beds. Omnidirectional antennas can be easily installed; however, severe interference restricts the available bit rate. Nodes must be properly separated in space and frequency domain to reduce interference. To this end, multichannel techniques 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 decreased; moreover, they provided an additional degree of freedom for assigning radio resources. Furthermore, directional antennas have a high antenna gain, which is useful for reducing the error rate and increasing channel bit rate. Therefore, using directional antennas is applicable to WMNs. In proposed approach, interference is neglected and directional antennas are used. In fact, even when using directional antennas, if the mesh routers are closely placed, interference may still appear. In this situation, interference can be eliminated using frequency diversity or high-gain antennas. In terms of cost, system complexity, and legal constraints, the restriction of the number of antennas accepted to be established in a mesh router is a node degree constraint on topology [30].

Although nodes in a WMN can operate in either time division duplex (TDD) or frequency division duplex (FDD) mode, TDD is selected here because the frequency assignment problem becomes simple, results in fewer required antennas, and has higher spectrum efficiency in the face of asymmetric traffic, such as HTTP and FTP. Each mesh router has accurate upstream and downstream traffic demands. Based on the flexibility of bandwidth allocation supplied 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 facilities the network model. The traffic demand of a mesh router consists of effective client traffic demands. Gateways connect to the Internet by wireline links. It is reasonable to suppose that the capacity of a wireline link is large; therefore, wireless links dominate the capacity and performance of a WMN. Gateway will show no traffic demand to the WMN because a gateway can directly supply its traffic demand by the wireline link. The capacity of a wireless link is limited by channel quality and path loss, and it suffers from attenuation and interference. Because channel quality changes with time, in planning a WMN, a link margin can be defined as an estimation of channel quality degradation. The model supposes the idea that link capacity can be fully utilized for data transmissions. Note that control overhead consumes some capacity, which decreases the effective capacity available for data transmission. Nevertheless, control overhead can be computed and deducted in advance so that only the effective capacity is considered during network topology optimization.

Now, the problem can be determined 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 declared 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:

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

such that

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

$$\sum_{v} a_{uv} \leq K \tag{C1}$$

$$c_{uv} a_{uv} \geq t_{uv} \tag{C2}$$

$$\left(\sum_{v} t_{uv} + \lambda_u \right) = \left(1 - \delta_{\mathrm{u}}\right) \sum_{\mathrm{v}} t_{vu} \tag{C3}$$

$$\sum_{u} \left(1 - \delta_u\right) \lambda_u = \sum_{u} \sum_{v} \delta_u\, t_{vu} \tag{C4}$$

$$a_{uv} - a_{vu} = 0 \tag{C5}$$

$$d_u \leq D \tag{C6}$$

The cost of setting up an antenna and setting mesh router *u* as a gateway is defined as 1 and σ_u, respectively. Constraint (C1) illustrates 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) defines traffic balance for each mesh router (non-gateway). Moreover, for a gateway mesh router, no traffic demand is present and a gateway does not submit traffic load to other mesh routers. Constraint (C4) requires that all traffic demands are assisted by gateways. Constraint (C5) means that a link is created by two opposite antennas. Constraint (C6) defines that the maximum packet delay of each mesh router is within an acceptable range. Furthermore, the WMN demands that each non-gateway mesh router should have at least two node-disjoint paths to different gateways, for the sake of survivability. Therefore, a WMN must satisfy the following survivability requirement:

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

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

4. GENETIC ALGORITHMS

A GA is a metaheuristic technique that is used to solve different optimization problems by imitating natural selection; i.e., the operation of adaptation to the environment carried out by living beings [31]. GAs are an attractive method to solving the complex problem summarized in the previous section. A GA determines a whole 'population' of 'individuals,' which are candidate solutions to the optimization problem. The distinguishing characterizes of each individual are coded into a structure called a 'chromosome'. The chromosome is a structure of genes, whose values can be selected from within a set of symbols. An application-dependant operation creates the individual by decoding its chromosome. The symbols employed as values of the genes are typically integer, real, or binary numbers, rely on the type of the problem. Once an individual is created, a fitness function is used to evaluate its fitness as a solution to the problem. Low values of fitness function are typically determined to the most fit individuals (minimization problem).

A GA begins with an initial population created either randomly or with some heuristic method that exploits the information of an expert in the problem area. The algorithm then advances in steps called generations. At each generation *g,* a new population $P(g + 1)$ is developed from $P(g)$. As generations pass, the population should globally improve on account of the application of genetic operators that imitate natural evolutionary techniques. To this end, the most fit individuals are chosen from $P(g)$ (selection) to be mated (crossover) and partially adjusted (mutation) so as to generate the new population $P(g + 1)$. The selection operation is used to determine which individuals in $P(g)$ should be selected to generate $P(g + 1)$. Optionally, an elite from among the chosen individuals (i.e., a small number of the best proceeding individuals) survives and is passed from $P(g)$ to $P(g + 1)$ without change. The crossover operation consists in choosing some of the individuals and mating them. In other words, it substitutes them with their children; i.e., individuals produced by mixing the genetic item in the parents' chromosomes. The real working of a crossover operation greatly depends on the encoding of the chromosome. Finally, the mutation operation presents some new genetic item in the population by randomly modifying the values of some genes. Many kinds of mutation operations can be identified to treat with different sets of symbols. The population continues to improve until a stopping criterion is achieved, with the simplest being a maximum number of generations. In addition, GA can be regarded as rapid steps for determining a 'good enough' solution to the problem; therefore they are interesting for practical problems [32]. GA is directly applicable, which allows them benefit with respect to the exact techniques. Therefore, a GA can be used to work in WMN design in which the solution may be iteratively updated.

5. SIMULATED ANNEALING

SA algorithm is a meta-heuristic planned for solving global optimization problems, i.e., finding a good vision to the global optimum of a function in a large search space. SA is imitated by the cooling process of metals by which a material is heated and then cooled in a controlled way to increase the size of its crystals and reduce their defects. The heat makes the atoms to depart their initial positions (a local minimum of energy) and move randomly; the slow cooling allows them more likelihood to discover configurations with lower energy than the previous one. In each iteration, it considers some neighbors of the current state *s*, and probabilistically decides between altering the system to the state *s'* or staying in the state *s*. The probabilities are selected so that the system meets towards lower energy states. Typically this step is repeated until a certain number of iterations is achieved or when the system reaches a state good enough for the application. The probability of making the transition to the new state *s'* is a function $P(\delta E, T)$ of the energy difference $\delta E = E(s') - E(s)$ between the two states, and the variable T , called temperature. The critical attributes of the SA algorithm is that the transition probability P is always non-zero, even when δE is positive, i.e., the system can move to a higher energy state (worse solution) than the current state. This fact permits the method to overcome local optima with probability $P=exp(-\delta E/T)$. So, when the temperature tends to a minimum, the probability tends to zero asymptotically. Thus, every time the algorithm accepts fewer moves to increase the system's energy. If δE is negative, i.e., the transition energy decreases, the movement is accepted with probability $P = 1$. The temperature decreases according to a particular function $T_{new} = €T_{old}$, where € is the parameter of cooling speed and €<1. The algorithm advances through the search space, [33, 34].

6. WMN TOPOLOGY DESIGN PROBLEM: TWO APPROACHES

We can now begin to illustrate how the GA and SA solve the WMN design problem. The inputs and definitions used in two proposed algorithms should be clarified. The locations of mesh routers and their traffic demands are given, the network configuration can be designed; the two algorithms are proposed to address the WMN design problem. A feasible network configuration (FNC) is a network configuration if, and only if, all constraints—i.e., from (C1) to (C7)—are satisfied. 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 route paths for traffic demands and the sequence to search a set of gateways such that the optimal solution (least cost) is acquired. Let *g* indicate 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 [30]. 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 progress to ensure if the network configuration meets that each mesh router must have at least two node-disjoint paths to different gateways which is called the survivability requirement (C7). 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 determined in the routing sequence. Dijkstra's algorithm [30] is used 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 suited 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 assigned. Links are assigned only when needed according to the path described by Dijkstra's algorithm. If a path passes through a potential link, then the potential link becomes assigned. The number of antennas used by the two end mesh routers of the given link is increased by one when a potential link is assigned. In this way, a network design that fulfils all traffic demands can finally be achieved. The paths for traffic demands are determined only in the design phase of the WMN. Mesh routers may choose the optimal paths for their traffic demands with respect to the current traffic load at runtime, quality-of-service requirement and specific metric.

6.1 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 well-chosen of parameters. These operations are detailed below:-

6.1.1 Encoding

Encoding is the basic process in a GA. Each mesh router is labelled a unique number ranging from 1 to N, where N is the number of mesh routers. In Fig. 1, the chromosome consists of two parts. The first part is the gateway indication, which uses a binary string to indicate whether the matching mesh router is a gateway. The second part of the chromosome is the sequence component, which uses integers ranging from 1 to N to denote the sequence for routing the path of the corresponding traffic demand. For example, a chromosome of 10010 12543 for a five-node network implies that mesh routers 1 and 4 are gateways, and it routes paths for the traffic demands of mesh routers 3, 2 and 5. Chromosomes are created by arbitrarily selecting gateways and sequence n

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

Fig 1: Chromosome encoding

6.1.2 Fitness function

A successful chromosome is the chromosome that produces an FNC; otherwise, it is a failed one. In the algorithm, the fitness value of a chromosome is calculated by *fitness = 1/Z*, where Z is the optimization function defined in Eq. 1. Therefore, a chromosome that corresponds to a greater fitness value will have a least-cost network design.

6.1.3 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.

6.1.4 Crossover

The common approach in crossover is single-point, whereby paired chromosomes are each cut at a randomly selected crossover place; the segments after the cuts are swapped to compose two new children chromosomes. Figure 2 illustrates the single-point crossover implementation.

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

Fig 2: Single-point crossover

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

6.1.5 Mutation

To run mutation operation, the chromosome replaces two randomly chosen genes in the gateway token and sequence parts, respectively, as described in Fig. 3.

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

Fig 3: Mutation process

The well choice of crossover probability P_c, as well as mutation probability P_m, is important to the GA. In Section 7, the study will search about the best values for them.

6.1.6 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:

- Generate randomly 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: Test for the best chromosome; if it does not have a successful chromosome go to Step 2.

Step 5: 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 6: Select candidate networks from the current population by means of the rank-selection method.

Step 7: Perform the crossover and mutation to obtain children candidate networks according to P_c and P_m, respectively.

Step 8: Establish the new population. Substitute the parents with children.

Step 9: Route the path for traffic demands and calculate fitness for each chromosome in the new population.

Step 10: Obtain the best chromosome of the new population.

Step 11: 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.

6.2 Simulated annealing approach

Among the possible heuristics that can solve optimization problem, the SA algorithm is chosen because it works iteratively keeping a single tentative solution S_a any time. In each iteration, a new solution S_n is created from the preceding one, S_a, and either replaces it or not depending on an acceptance measure. The acceptance measure operates as follows: both the new (S_n) and old (S_a) solutions have an associated quality value, determined by an objective function (also called fitness function). If this new solution is better than the old, then it will substitute it. If it is worse, it substitutes it with probability P. As iterations continue, the temperature parameter value is decreased following a cooling schedule, thus biasing SA towards accepting only better solutions. The steps of the overall algorithm for SA to solve the WMN design problem are as follows [33]:

Step 1: Set the parameters. Set initial temperature (T_0), final temperature (T_f), cooling speed parameter €, 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 (S_0) that represents a network similar to the chromosome structure in the GA method.

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

Fig 4: Solution in SA

Step 4: Route the path for traffic demands and calculate fitness for current solution S_0.

Step 5: Check the current solution S_0; if it does not have a successful solution, go to Step 2.

Step 6: Pick random solution S from neighbourhood solutions of S_0. The mutation method in the GA can be used to produce neighbourhood solutions. For example, one of the neighbourhood solutions can be represented as shown in Fig. 5.

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

Fig 5: Neighborhood solution

Step 7: Route the path for traffic demands and calculate fitness for selected neighbourhood solution S.

Step 8: Check the solution S; if it does not have a successful solution, go to Step 6.

Step 9: Compute the difference between two solutions, $\delta f = f(S) - f(S_0)$.

Step 10: Check acceptance condition, if $\delta f < 0$, set S as current solution S_0; otherwise generate random number *rand* between 0 and 1, test if *rand* < exp($-\delta f/T$), set S as current solution S_0.

Step 11: Assign solution S_0 as the best solution and compare the current best solution with the best solution from the previous. If it is better, it replaces the best solution.

Step 12: Perform the iteration test. If $I < maxit$, set $I = I + 1$, and proceed to Step 6 for the next iteration.

Step 13: Update temperature $T_{new} = € T_{old}$.

Step 14: Check $T_{new} > T_f$, if it was achieved; set $I=1$ then go to Step 6; 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.

7. COMPUTATIONAL RESULTS

This paper performed the application in Visual C++ 2010. The program 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 1. The performance of the two proposed algorithms is evaluated by using three test cases that represent different network configurations. First, GA and SA parameters values are presented to serve as a comparison study in later of this section. The GA parameters are *Pop_size* = 20, P_c = 0.4, and P_m = 0.4. The SA parameters are T_0 = 100, T_f = 0.01 and ϵ = 0.5. Two algorithms are executed at a different number of iterations for each test case.

Table 1. Experimental results

Network	Iteration number	Comparison	GA	SA
20-mesh routers	10	Gateway	2	2
		Cost(units)	548	576
		Number of antennas	92	111
		Best iteration	7	4 (T=1.5625)
		Time(ms)	1890	1479
	50	Gateway	2	2
		Cost(units)	542	556
		Number of antennas	95	113
		Best iteration	34	30 (T=0.0976)
		Time(ms)	9053	6977
	100	Gateway	2	2
		Cost(units)	536	552
		Number of antennas	92	102
		Best iteration	56	51 (T=0.0976)
		Time(ms)	17589	13869
50-mesh routers	10	Gateway	2	2
		Cost(units)	963	852
		Number of antennas	370	410
		Best iteration	8	9 (T=0.0122)
		Time(ms)	24364	20793
	50	Gateway	2	2
		Cost(units)	824	841
		Number of antennas	342	402
		Best iteration	50	40 (T=0.0976)
		Time(ms)	138021	103667
	100	Gateway	2	2
		Cost(units)	787	828
		Number of antennas	346	386
		Best iteration	67	40 (T=0.0122)
		Time(ms)	262934	211395
100-mesh routers	10	Gateway	3	3
		Cost(units)	1745	1354
		Number of antennas	390	396
		Best iteration	4	4 (T=25)

		Time(ms)	122916	89294
	50	Gateway	2	2
		Cost(units)	933	845
		Number of antennas	386	392
		Best iteration	42	25 (T=3.125)
		Time(ms)	672359	312718
	100	Gateway	2	2
		Cost(units)	895	838
		Number of antennas	386	390
		Best iteration	45	42 (T=12.5)
		Time(ms)	1.17e+006	681723

Important observations can be concluded from the results shown in Table 1:

- An increase in the number of iterations tends to an improvement (minimizing of) network costs in GA and SA.
- In small-size networks, the GA is better than SA for minimizing network costs. However, in large-size networks, SA is better than the GA.
- SA takes short running time in all test cases comparing with GA.
- Number of antennas is one of the major impacts in WMNs design; GA has the least used antennas.
- In large size network, more number of gateways is needed to find minimum costs at small number of iterations.
- GA accelerates to find best costs at all number of iterations.

Because the GA and SA are based on randomization in their operations, an insignificant number of solutions that is not compatible with previous search can be found. The study advises to use GA with a small-size network and use SA with large-size network. If running time is desired as a first priority factor in solving the optimization problem, then SA is the first choice.

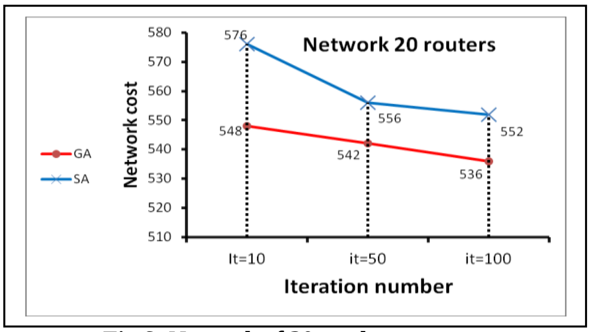

Fig 6: Network of 20 mesh routers cost

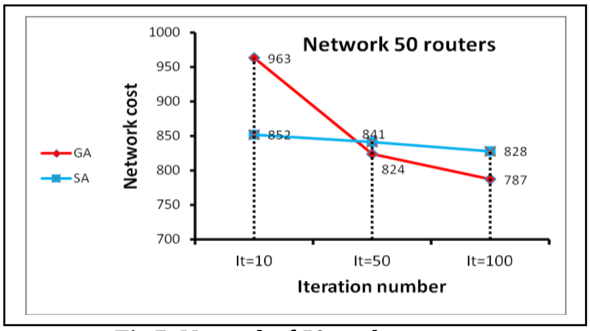

Fig 7: Network of 50 mesh routers cost

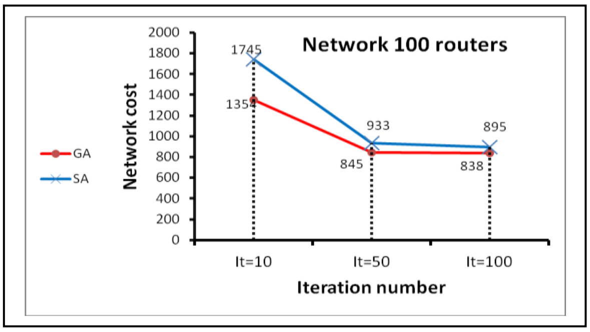

Fig 8: Network of 100 mesh routers cost

7.1 Optimization with GA parameters

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

7.1.1 Probability of crossover

Probability of crossover specifies the rate of crossover (mating) revolving between two chromosomes. The values of P_c are varied to find the best value for the optimization problem (to minimize cost); the best value is found when P_c =0.4, as shown in Table 2. The values for other parameters are (*iteration number* = 100, *Pop_size* = 20, and P_m = 0.4) with a 20-mesh routers network.

Table 2. Crossover probability optimization

P_c	Cost (units)	P_c	Cost (units)
0.1	548	0.6	554
0.2	544	0.7	538
0.3	550	0.8	540
0.4	536	0.9	544
0.5	544		

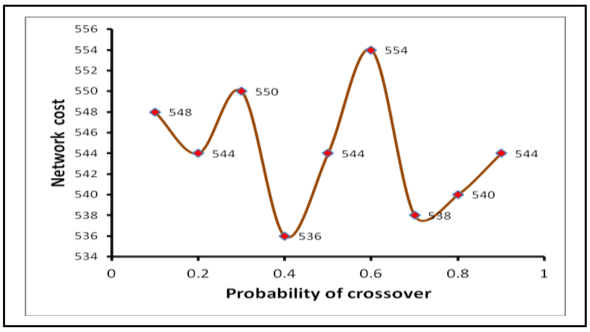

Fig 9: Cost – probability of crossover

7.1.2 Probability of mutation

Probability of mutation identifies how often parts of chromosomes will be mutated. The values of P_m are varied to find the best value for the optimization problem (to minimize cost); the best value is found when P_m = 0.9, as shown in Table 3. The values for other parameters are (*iteration number* = 100, *Pop_size* = 20 and P_c = 0.4) with a 20-mesh routers network.

Table 3. Mutation probability optimization

P_m	Cost (units)	P_m	Cost (units)
0.1	546	0.6	538
0.2	540	0.7	544
0.3	538	0.8	540
0.4	544	0.9	534
0.5	538		

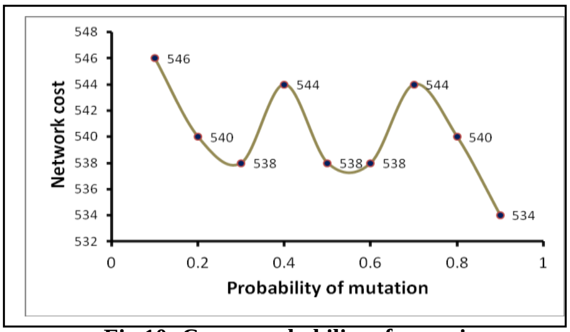

Fig 10: Cost – probability of mutation

7.1.3 Population size

Population size assigned how many chromosomes are available and, therefore, how much genetic material is available for operation during the search. The values of population size of the generation are varied to find the best value for the optimization problem (to minimize cost); the best value is found when *Pop_size* = 40, as shown in Table 4. The values for other parameters are (*iteration number* = 100, P_m = 0.9 and P_c = 0.4) with a 20-mesh routers network.

Table 4. Population size optimization

Population size	Cost (units)	Population size	Cost (units)
10	544	60	534
20	542	70	534
30	538	80	532
40	530	90	536
50	532	100	534

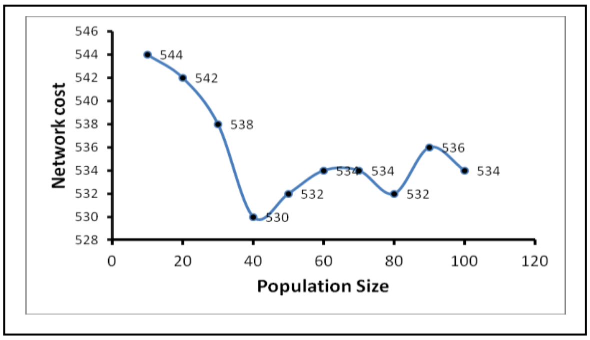

Fig 11: Cost – population size

7.2 Optimization with SA parameters

In SA, experiments are presented to examine the relationship between SA parameters (initial temperature, final temperature and cooling speed) and the optimization problem (to minimize cost).

7.2.1 Initial temperature

The values of initial temperature *(T_0)* are changed to find the best value for the optimization problem (to minimize cost); best value is found when T_0 = 400, 800 and 1000, as shown in Table 5. The values for other parameters are (*iteration number* = 100, *final temperature*=0.01, and *cooling speed* = 0.5) with a 20-mesh routers network.

Table 5. Initial temperature optimization

T_0	Cost (units)	T_0	Cost (units)
100	552	600	548
200	550	700	548
300	550	800	546
400	546	900	550
500	552	1000	546

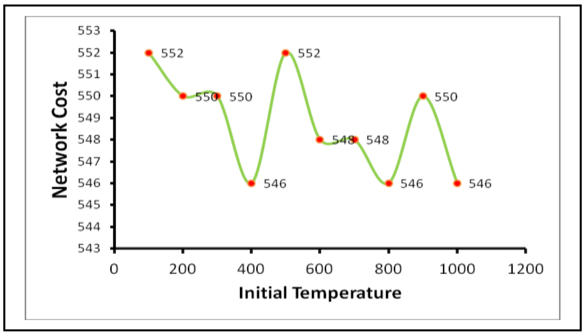

Fig 12: Cost – initial temperature

7.2.2 Final temperature

The values of final temperature *(T_f)* are changed to find the best value for the optimization problem (to minimize cost); the best value is found when T_f=0.07, as shown in Table 6. The values for other parameters are (*iteration number* = 100, *initial temperature*=800, and *cooling speed* = 0.5) with a 20-mesh routers network.

Table 6. Final temperature optimization

T_f	Cost (units)	T_f	Cost (units)
0.01	552	0.06	554
0.02	556	0.07	546
0.03	552	0.08	552
0.04	554	0.09	552
0.05	552	0.1	558

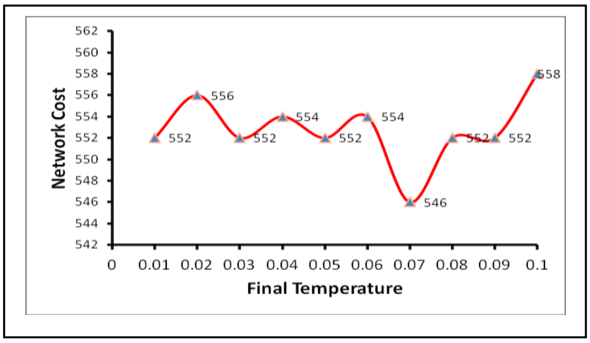

Fig 13: Cost – final temperature

7.2.3 Cooling speed

Cooling speed parameter is a temperature reduced ration in SA. The values of cooling speed *(€)* are changed to find the best value for the optimization problem (to minimize cost); the best value is found when €=0.7, as shown in Table 7. The values for other parameters are (iteration number = 100, initial temperature=800, and final temperature = 0.07) with a 20-mesh routers network.

Table 7. Cooling speed optimization

€	Cost (units)	€	Cost (units)
0.1	554	0. 6	560
0. 2	552	0. 7	548
0. 3	558	0. 8	552
0. 4	552	0. 9	550
0. 5	552		

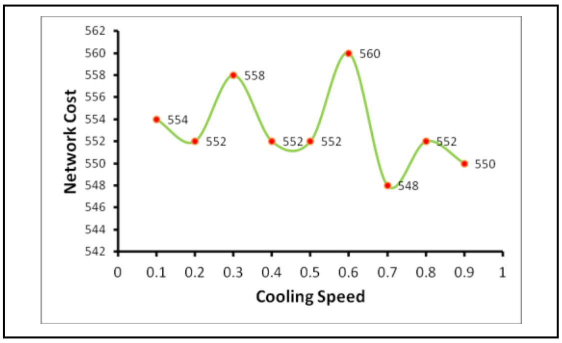

Fig 14: Cost – cooling speed

The search result for the best values to each parameter in GA and SA algorithms is changeful because two algorithms rely on random selection in its operation. The determined best values may be varied when execution program again.

8. CONCLUSIONS AND FUTURE WORK

In this paper, a GA and SA are presented for the WMN design problem. The functional objective was to minimize cost and search gateways of the WMN under constraints. Performance comparisons of the GA and SA under different networks were presented. In addition, the running time required for these algorithms was evaluated. The results conclusion is the GA and SA were able to minimize cost. The GA was better than SA in a small-size network; however, SA was better in a large-size network. Further, the optimal value for parameters in the GA and SA was examined. In future work, we intend to propose another metaheuristic method to further advance the WMN design problem.

9. 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] S. Kirkpatrik, C. D. Gelatt and M. P. Vecchi, Optimization by simulated annealing, science, vol. 220, pp. 671-680, 1983.

[7] I. Bohachevsky, M. E. Johnson and M. L. Stein, Generalized simulated annealing for function optimization, in Technometrics, vol. 28, pp.209-217, 1986.

[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] C. Ying-Yu, L. Chen-Chun and C. Chien, Channel assignment and routing for multi-channel wireless mesh networks using simulated annealing, Global Communications Conference GLOBECOM, vol. 27, pp. 1-5, 2006.

[29] F. Xhafa, C. Sanchez, L. Barolli and R. Miho, An annealing to router nodes placement problem in wireless mesh networks, International Conference on Complex Intelligent and Software Intensive Systems, pp. 245-252, 2010.

[30] 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.

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

[32] 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.

[33] S. Sakamoto, E. Kulla, T. Oda, M. Ikeda, L. Barolli and F. Xhafa, A comparison study of simulated annealing and genetic algorithm for node placement problem in wireless mesh networks, Journal of Mobil Multimedia, vol. 9, pp.101-110, 2013

[34] 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.