Optimizing the Placement of ITAPs in Wireless Mesh Networks by Implementing HC and SA Algorithms

Liqaa Nawaf, Christine Mumford, and Stuart Allen

Cardiff University, Computer Science, Cardiff, UK
`{NawafLF,MumfordCL,AllenSM}@cardiff.ac.uk`

Abstract. In this paper, we present novel heuristic improvement (move) operators for the design of Wireless Mesh Networks (WMN), and demonstrate their efficiency within simple Hill Climbing (HC) and Simulated Annealing (SA) frameworks. The management cost of Internet Transit Access Points (ITAPs) in WMN is significant, so it is crucial to minimize the number of ITAPs required whilst maintaining an acceptable quality of service (QoS). Using a single objective method, we investigate algorithms to make informed placement decisions based on the grid size, wireless range connectivity, wireless link capacity and user demands. The experimental results showed the efficiency of the proposed combination of move operators.

Keywords: wireless mesh network, move operators, optimization.

1 Introduction

Wireless Mesh Networks (WMNs) are a promising approach to provide ubiquitous broadband internet access due to their great potential in supporting multimedia applications. WMNs, an emerging technology, may bring the dream of a seamlessly connected world into reality. In a WMN, a limited number of Internet Transit Access Points (ITAPs) serve as gateways or bridges to the Internet, and are deployed across a community. Individual houses within the community are equipped with antennae and low cost routers (namely, mesh routers) which perform two roles: 1) to service the traffic in and out of the house to the individual laptops and other devices (that is, to the mesh clients), and 2) to provide a link in a multi-hop wireless backbone that is formed between the houses to cooperatively route traffic throughout the neighbourhood, communicating with the Internet through the ITAPs. Such a multi-hop structure dramatically reduces the number of ITAPs needed, which can result in massive savings in cost. The networking infrastructure is decentralized and simplified because each node need only transmit as far as the next node. An ITAP will share its Internet connection wirelessly with all houses in its vicinity. Those houses then share the connection wirelessly with the nodes closest to them. Large or small networks can be created in this way to serve small rural communities, or millions of residents in a city. In this paper, we address the ITAP placement problem, which involves determining the numbers and locations of ITAPs that are required to service the needs of a community. This situation is easily modelled using graph theory. In wireless

N. Mitton et al. (Eds.): AdHocNets 2015, LNICST 155, pp. 29–41, 2015.
Doi: 10.1007/978-3-319-25067-0_3

neighbourhood networks, a set of Houses and set of ITAPs are designed and deployed. Fixed capacities are associated with each house and ITAP and with all connecting edges in the network.

The focus of our present paper is to demonstrate the effectiveness of our novel move operators on a simple ideal link model with a single objective. Our simple ideal link model is taken from [3]. To the best of our knowledge, our proposed move operators have not previously been applied to WMNs. We show that, by using a suitable combination of move operators to place the ITAPs, rapid improvements can be made to the underlying WMN. The remainder of this paper is structured as follows: we discuss related work in section 2. Section 3 outlines the ideal link model, and section 4 formulates the associated integer linear program. In section 5, our heuristic algorithms are described, and the experimental methodology and results are presented in section 6. Finally, conclusions are drawn in section 7.

2 Related Work

[1] Addresses the problem of ITAP (gateway) placement, consisting of placing a minimum number of ITAPs so that Quality of Service (QoS) requirements are satisfied. In a wireless mesh network, traffic is aggregated and forwarded towards the ITAPs. The authors formulate the ITAP placement problem given by integer linear programming (ILP) and show that the problem of finding a minimum number of ITAPs is NP-Hard. In practice, an LP solver, such as CPLEX, can only handle small sized networks under the proposed model due to the rapid increase in the number of variables and constraints with increasing network size. The authors proposed an algorithm that produces recursive approximations of the Minimum Dominating Set problem. When compared to other algorithms, their approach reduced the number of ITAPs by up to 50%, whilst at the same time exhibiting smooth and consistent performance when subject to different QoS constraints. From [1] it is clear that CPLEX is too slow to use on large instances, and some experimental work we undertook (not reported here) also supports this conclusion. In [2]the fundamental issue is the placement of mesh routers in a local network that has one ITAP, and to find the minimal configuration of mesh routers so as to satisfy the network coverage, connectivity, and Internet traffic demand. The rate of the increase with respect to the number of ITAPs increases with the network size, which indicates that the traffic density has significant impact on the number of required ITAPs in the larger size of networks. Chandra et al. [3] developed algorithms to place ITAPs in multi-hop wireless network to minimize the number of ITAPs while satisfying users' bandwidth requirements. They formed the ITAP placement problems as linear programs and presented several greedy-based approximation algorithms. The placement of ITAPs is based on neighbourhood layout, user demand, and wireless link characteristics. In [4] the authors observe that the transmission rate and the channel utilization required to satisfy the WMN's demand clearly depend on device technologies, but in particular

on the distance between the mesh client and the ITAPs to which it is connected; hence, the allocation mechanism influences the number of mesh clients that have the opportunity to exploit the available bandwidth. The authors in [5] minimize the network installation cost while providing full coverage to wireless mesh clients. In this type of network architecture, a limited number of ITAPs wirelessly connected to the wired realm can potentially provide low cost internet connectivity to a large number of mesh routers. The authors of [5] and [6], also supported by our experimental work, indicate that the heuristics can provide close to optimal solutions even for large instances. Jangeun Jun and Mihail L. Sichitu [7] discuss the problem of the capacity of WMNs and the fairness of multi-hop networks; for example, as more nodes are installed, the reliability and network coverage increase.

The obtained numerical results of [8] show that directional antennas can greatly enhance the performance of WMNs, thus enabling high throughput services. The cause of the problems of wireless network performance can be traced back to the original design assumptions. The authors [9] believe that a well-planned and optimized wireless network can often provide extra capacity with the same infrastructure cost; for instance, this may result from more efficient use of radio frequencies. Another way to achieve a better network performance is to optimize the placement and characteristics of ITAPs before network deployment. A careful placement of ITAPs, may lead to less congestion, low delay and eventually better throughput if the distances and the links capacity are taken into account. The authors [10] considered neighbourhood search-based methods as more powerful methods than ad hoc methods for achieving near optimal placements of mesh router nodes. Their experimental evaluation demonstrated an excellent performance of a swap-based movement neighbourhood search. In [11] the neighbourhoods are based on reversing segments of routes (sub-routes) and exchanging segments between routes. The authors demonstrated that their Variable Neighbourhood Search is very competitive compared with previously published state-of-the-art heuristic algorithms. In [12] the authors considered three different types of movements (Random, Radius and Swap). Their results showed a very good performance of combination movement for Router Nodes placement problem in WMNs. In [13] they propose a grid-based ITAPs (called Gateways there) deployment method using a cross-layer throughput optimization, and prove that the achieved throughput using this method is a constant times of the optimal.

From the literature review, we observe that local search algorithms with suitable heuristic move operators can produce acceptable results on instances that are too large for solutions by exact methods. In this paper, we propose a new two phase approach for solving the network optimization problem, in which the underlying framework iterates between the two phases. The first phase focusses on improving the QoS that can be obtained from deploying a fixed number of ITAPs by moving individual ITAPs from one place to another. The second phase then attempts to reduce the number of ITAPs required, whilst still maintaining adequate QoS. To effectively support different goals of the two phases, a mix of operators is required. In our experimental work, we demonstrate the effectiveness of combining our new operators in suitable proportions. We compare hill climbing (HC) and simulated annealing (SA) as heuristic drivers for two phase approach.

3 Network Model

We use the ideal link model proposed in [3], and aim to minimize the number of required ITAPs, while satisfying users' bandwidth requirements. Our algorithms are applied to place ITAPs in different locations, and the network flow algorithm of Ford-Fulkerson [14] is then used to ensure that demand is satisfied for a specific WMN configuration. Before the application of Ford-Fulkerson, however, it is necessary to transform the WMN into an equivalent single source, single sink model. Once this has been done, the Ford-Fulkerson Algorithm can be used to compute the maximum flow capacity of the edges in the network flow, and hence whether not or a given WMN configuration is able to support the required user demand.

In more detail, a set of houses $H=\{h_1,\ldots h_n\}$ is given in the network, along with a set (I) of locations at which ITAPs can be installed. Each house has a traffic demand, wh, and we say a house is served if this traffic can be successfully transmitted to an active ITAP (possibly by a sequence of hops). Traffic from each house can be routed along multiple paths simultaneously, hence the problem can be modelled using max flow algorithms. We construct a graph with $H \cup I$ as nodes, with edges joining each pair that are within wireless range. The capacity of each edge, Cap_e, is the data rate that can be sustained on that link, and each node has a capacity Cap_h , or $Cap_h = Cap_e$, which denotes their ability to process and forward data. Each ITAP also has a capacity limit, based on its connection to the Internet and its processing speed, where the Cap_i denotes the capacity of ITAP. To complete the graph, we add a source (joined by edges of capacity Cap_e to each house) and a sink (joined by edges of capacity Cap_e to each ITAP).

4 Integer Linear Program Formulation

In this section, we describe the ideal link model of [3]; it is defined by the equations and inequalities listed below. For each edge e and house h, we have a variable $x_{e,h}$ to indicate the amount of flow from h to the ITAPs that is routed through e. For each $ITAP_i$,, a variable y_i indicates the number of ITAPs opened at the location i (more precisely, y_i is the number of ITAPs opened at locations in the equivalence class i). As defined previously in the present paper, Cap_e, Cap_h , and Cap_i denote the capacity of the edge e, house h, and $ITAP_i$, respectively; wh denotes the traffic demand generated from house h.

The first constraint, equation (1) formulates the flow conservation constraint, namely, for every house except the house originating the flow, the total amount of flow entering the house is equal to the total amount of flow exiting it. The inequality, equation (2) formulates the constraint that each house has wh amount of flow to send, and the third constraint indicates that a house does not receive flow sent by itself. The next three inequalities of the integer program capture the capacity constraints on the edges, houses, and ITAPs. Equation (7) says that no house is allowed to send any traffic to an ITAP unless the ITAP is open. Notice that this inequality is redundant and follows from the ITAP capacity constraint and the assumption that y_i is an integer.

Minimize $\sum_{i \in x} y_i$

Subject

$$\sum_{e=(v,h')} x_{e,h} = \sum_{e=(h',v)} x_{e,h} \quad \forall h, h' \in H,\ h' \neq h \tag{1}$$

$$\sum_{e=(h,v)} x_{e,h} \geq wh \quad \forall h \in H \tag{2}$$

$$\sum_{e=(v,h)} x_{e,h} = 0 \quad \forall h \in H \tag{3}$$

$$\sum_{h} x_{e,h} \leq Cap_e \quad \forall e \in E(G) \tag{4}$$

$$\sum_{h',e=(v,h)} x_{e,h'} \leq Cap_h \quad \forall h \in H \tag{5}$$

$$\sum_{h',e=(v,i)} x_{e,h'} \leq Cap_i\, y_i \quad \forall i \in I \tag{6}$$

$$\sum_{e=(v,i)} x_{e,h} \leq wh\, y_i \quad \forall i \in I,\ h \in H \tag{7}$$

$$x_{e,h} \geq 0 \quad \forall e \in E(G), h \in H \tag{8}$$

$$y_i \in \{0,1,2,\ldots\} \quad \forall i \in I$$

To generate the maximum flow of edges, a graph (G) is constructed consisting of nodes and edges (X, Y) between them. Every house node is directly connected to the source node and every ITAP node is directly connected to the sink node. Let graph Gp be the sub-graph of G with the same capacities as G.
House *h* node;

- For each house *h* in graph G replace it with two nodes *h_in* and *h_out* and connect those nodes using directed edges.
- Add edges between *h_in* and *h_out* nodes to graph Gp.
- Source is connected directly to every (*h_in*) in the Graph.

Edge (X, Y) in G;

- Add edge (*X* , *Y) to* graph Gp.
- Add some capacities to the edges.

ITAPi in G;

- For each ITAP in graph G replace it with two nodes *ITAP_in* and *ITAP_out* and connect those nodes using directed edges.
- Add edges between *ITAP_in* and *ITAP_out* nodes to graph Gp.
- Sink is connected to each ITAP.
- The connection of edges for the graph Gp would be starting from the source node and go through *h_in*, *h_out*, *ITAP_in* and *ITAP_out* then to the sink node, to compute the maximum flow of the network, as shown below in figure 1.

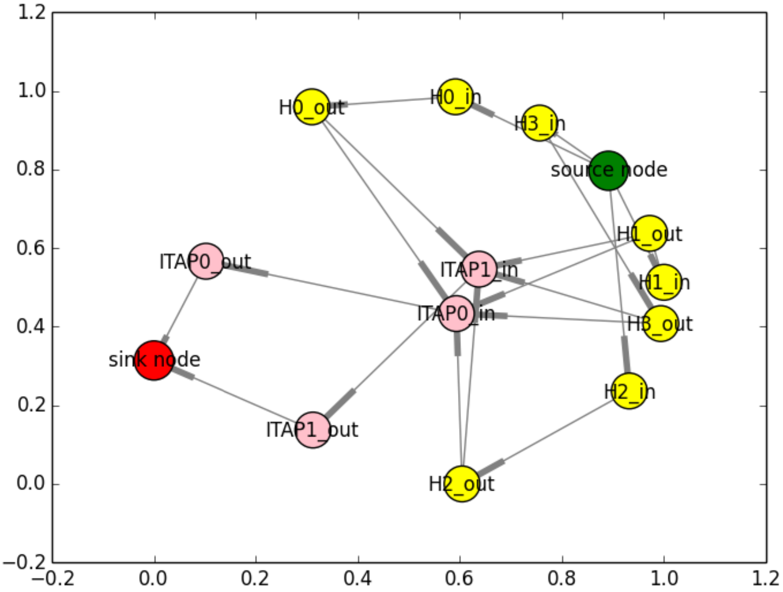

Fig. 1. The model of 4 houses and 2 ITAPs with source and sink.

5 Our Heuristic Algorithms

We experiment with Hill Climbing and Simulated Annealing algorithms to provide simple frameworks for experimenting with our proposed move operators: Swap, Reallocate, Delete and Add. Each one of these moves is designed to work on different aspects of the candidate WMN configuration, and our goal is to use combinations of our moves to sample the search space efficiently and effectively. The "Swap" exchanges the numbers of ITAPs from two locations, while "Reallocate" moves a single ITAP from one location to another. The "Delete" removes an ITAP from a given location, and "Add" increments the number of ITAPs at a given location.

Our approach to WMN design requires two phases: 1) Optimizing the placement of a fixed number of ITAPs to maximize the network flow, and 2) Using the excess network capacity provided by Phase 1, minimizing the number of ITAPs. In the first phase, we employ only the reallocate and swap move, which does not change the number of ITAPs, but can relocate them to give a better value of maximum flow, following the application of the Ford-Fulkerson algorithm. The second phase is aimed at minimizing the number of ITAPs, which is the main objective. In the second phase, we use all four of our moves.

We represent a solution by a vector $s_1, s_2, \ldots, s_n$ where s_i is the number of ITAPs to be installed at location $_i$. A move modifies a solution s to a new solution s' as follows;

- Swap; select i and j at random with i≠j, and set
  s'[i], s'[j] =s[j], s[i]

- Reallocate; select i and j at random with i≠j and s[i]>0, and set
  s'[i] = s'[i] -1
  s'[j] = s'[j] +1
- Delete; select j at random with s[j]>0, and set
  s'[j] = s[j] -1
- Add; select j at random, and set
  s'[j] = s[j] +1

5.1 Hill Climbing

Hill climbing (HC) starts with a sub-optimal solution to a problem (that is, starting at the base of a hill) and then attempts to repeatedly improve the solution in small steps (neighbourhood moves) until some condition is maximized (the top of the hill is reached). HC Algorithm will simply accept neighbour solutions that are better than the current solution; when HC cannot find any better neighbours, it stops. HC usually does not find a global optimum; more likely, it gets stuck in a local optimum.

5.2 Simulated Annealing

Simulated Annealing (SA) is similar to HC, but a little more sophisticated. It has been used successfully in solving many combinatorial optimization problems, and is better at avoiding local optima than HC if it is well implemented. SA was originally inspired by the slow cooling of metals to form crystalline structures of minimum energy, and Metropolis et al. (1953)[15] first introduced these principles into numerical minimization. Like HC, SA is launched with a starting configuration, s_0, and then works through a large number of neighbourhood moves (s to s') in an attempt to produce better solutions. Unlike HC, however, the acceptance criterion for s' is less strict, allowing the algorithm to jump out of local optima. If the new solution, s', is better than our old solution, s, it is accepted unconditionally. If, however, the new solution is worse, then it is accepted with a certain probability, related firstly to how much worse it is; and secondly, to how high the current "temperature" of our system is. At high temperatures, the system is more likely to accept solutions that are worse. In practice, a simulated annealing implementation is typically constructed within two nested loops. In the outer loop, the temperature is reduced gradually, and within the inner loop many perturbations of s to s' are tried. At each step, s' faces an acceptance test, based on $\Delta = \mathrm{Cost}(s') - \mathrm{Cost}(s)$. The new solution, s', is accepted with a probability of 1, if $\mathrm{Cost}(s') < \mathrm{Cost}(s)$ (just like HC), and with a probability of $e^{-\Delta/T}$ otherwise, where T is the current temperature. In our system, we set the initial temperature, $T_0 = 2$.

5.3 Implementation of HC and SA in the Ideal Link Model

We evaluate the performance of HC and SA Algorithm using various levels of wireless range connectivity in the context of maximizing throughput and reducing the number of ITAPs. In Figure 2, we provide a framework that covers our HC and SA algorithms.

```
Algorithm 1. Framework for our Hill Climbing and Simulated Annealing Approaches

Our objective function is to maximize the throughput in Phase 1, and to minimize
the number of ITAPs in Phase 2.
    Generate an initial solution $s$ randomly
    {Set initial Temperature, $T_0$–SA only}
    {While $T > T_{min}$ Do – SA only}
        While number of iterations < max iterations Do
                Generate a new solution $s'$ within the neighbourhood of $s(s' \in N(s))$
                $\Delta \leftarrow Cost(s') - Cost(s)$
                If acceptance criterion is passed (see text) for HC or SA, $s \leftarrow s'$

Phase 1:
            For Each possible ITAP Location
                Select one move randomly
                Apply Swap OR Reallocate move to the ITAPs
                Evaluate the amount the throughput
            End for
Phase 2:
            For Each possible ITAP Location
                Select one move randomly
                Apply Swap OR Reallocate OR Delete OR Add move to the ITAPs
                Evaluate the number of unsatisfied houses and number of ITAPs
            End For
        End While
    {Reduce T – SA only}
{End While – SA only}
        Return $s$
```

Fig. 2. General Pseudo Code Outline of our Heuristic Methods

The cost function plays a key role in any optimization problem. It is through its calculation that one can measure the quality of any solution. Hence, its correct definition is essential for the behaviour of any search algorithm. Our aim is to minimize the required number of ITAPs, whilst still maintaining adequate QoS which maximize the number of houses served. To formulate this using a single cost function, we instead consider minimizing a weighted sum of the number of ITAPs and the unserved houses.

$$E(s) = Fw * \sum_{i \in h} Wh - \sum_{i \in I} F + Iw \sum_{i \in x} S \qquad (9)$$

In equation (9), variable *E(s)* which represents the cost function; *Fw* indicates the flow weight of the house (more precisely, *Fw* represents the flow cost of each house). *Wh* indicates the traffic demand generated from house *h*, and let *F* denote the amount of traffic routed to ITAP$\grave{\imath}$ in this solution where $\grave{\imath} \in$ multiset of ITAPs, we have a variable *Iw* that indicate the weight of the ITAPs. For each ITAP*i* we have a variable *S* which represents the solution (number of ITAPs opened at locations *i*) *S* may be written: $S = (s_1, s_2, ..., s_n)$ for some variables s_i.

6 Experimental Methodology and Results

The initial temperature of the SA is (T_0= 2) and then the temperature is gradually reduced using a cooling schedule. To test our approach effectively, we generated a wide range of instances for our experiments, based on a random uniform distribution of nodes. We varied the number of houses, the number of ITAPs, the number of ITAP locations and the grid size. For instances with the same number of houses, those houses are placed at the same locations for each instance. Each experiment, runs five ranges three times and reports the average and best result with the runtime as shown in tables below. For experiments 1-10, only Phase 1 is applied, which attempts to maximize the throughput (traffic flow) by moving ITAPS from one place to another, without changing the number of ITAPs. The aim is to demonstrate the effectiveness of the two moves in improving the throughput for a range of instances with different characteristics (such as ITAP capacity and number of ITAP locations, etc.). For these experiments, only the swap and reallocate move are needed. For experiments 11-14, we apply both phases of the HC and SA algorithms, using all four move operators. The main goal here is to minimize the number of ITAPs.

6.1 Performance and Evaluation

We illustrate the numerical results obtained to minimize the number of ITAPs using combinations of different move operators in the HC and SA algorithms. In tables 1–6. Table 1, shows a comparison of the individual move operators, such as Swap move and Reallocate move, and we can observe that the reallocate move outperforms the swap move but takes longer. Nevertheless, using swap move reduces the chance of being trapped in local optimum [16], hence both moves are used together to improve the throughput and running time, as shown in Table 2. It can be observed from the results that there is improvement in the throughput when the number of ITAP locations and wireless range connectivity are increased. It is important to note that the throughput probability is determined by the WRC, distribution number of ITAP Location and the network size, as shown in Table 2. We find that HC works well in small and simple instances (in table 2, EXP. 1, 2, 8 and 9) but does not work as well as the SA on large instances (EXP. 3-7 and 10). The results for experiments 11 – 14 in Tables (3 – 6) show that less ITAPs are required to satisfy the demands using the first and second phases together. However, the cost function that we use for these experiments shows that there is a trade-off between minimizing ITAPs and unsatisfied houses. Table 3 illustrates our evaluation of the combination of move operators with the SA algorithm as good and efficient in satisfying the houses demand with less ITAPs, which is near to the optimal solutions as shown in Figure 3 for experiment 12. Thus, the cost function in SA surpasses HC and yields excellent solutions by effective combinations of different move operators, as shown in Tables (3 and 5). However, in Tables (4 and 6), using swap and reallocate move only demonstrates no improvement in minimizing the number of ITAPs.

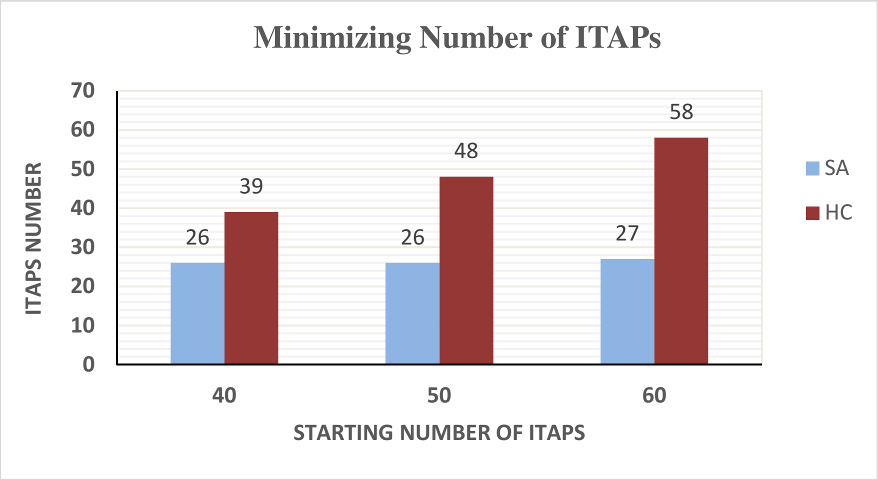

Fig. 3. Result of experiment 12 starting with different number of ITAPs (40, 50 and 60)

Table 1. Throughput value for Reaallocate and Swap move individually

EXP	#House	#ITAP	#ITAP-Location	ITAP-Cap	Grid-Area	WRC	W-Link	#Iterations	Cost-Real HC(RT/sec)	Cost-Swap HC(RT/sec)	Cost-Real SA(RT/sec)	Cost-Swap SA(RT/sec)
1	100	10	10	10	100*100	25	5	500	100(7)	100(3)	100(69)	100(67)
3	500	50	50	20	500*500	25	20	500	360(625)	348(615)	366(669)	365(663)
5	500	50	50	20	500*500	25	15	1000	364(1289)	354(1265)	366(1356)	366(1345)
8	500	50	100	20	500*500	30	15	500	495(1071)	494(1040)	495(1073)	495(1060)
10	500	50	100	20	1000*1000	50	15	500	434(892)	430(866)	469(935)	461(922)

Table 2. Throughput value and running time of Experiments 1- 10

EXP	#House	#ITAP	#ITAP-Location	ITAP-Cap	Grid-Area	WRC	W-Link	#Iterations	Cost HC(RT/sec)	Cost SA(RT/sec)
1	100	10	10	10	100*100	25	5	500	100(6)	100(66)
2	500	50	10	10	100*100	25	5	1000	500(173)	500(3623)
3	500	50	50	20	500*500	25	20	500	360(623)	366(665)
4	500	50	70	20	500*500	25	15	500	395(762)	402(797)
5	500	50	50	20	500*500	25	15	1000	362(1267)	366(1332)
6	500	50	100	20	500*500	25	20	500	431(862)	466(935)
7	500	50	100	20	500*500	25	15	1000	440(1712)	467(1852)
8	500	50	100	20	500*500	30	15	500	495(1027)	495(1045)
9	500	50	50	20	500*500	35	15	500	500(1000)	500(1029)
10	500	50	100	20	1000*1000	50	15	500	432(841)	463(889)

Table 3. Cost function of Experiment (11-14) using all move operators with different number of ITAPs in SA Algorithm.

EXP	#House	#ITAP-Location	ITAP-Cap	Grid-Area	WRC	W-Link	#Iterations	Cost-min(Avg)	RT/sec	Satisfied House	#ITAP End(start)
11	500	10	10	100*100	25	5	500	500(500)	1759	500	50(50)
12	500	100	20	500*500	30	15	1000	415(425)	2152	495	26(50)
13	500	50	20	500*500	35	15	3000	250(250	6099	500	25(50)
14	1000	100	20	500*500	35	15	1000	500(500)	9006	1000	50(50)
11	500	10	10	100*100	25	5	500	500(500)	1755	500	50(40)
12	500	100	20	500*500	30	15	1000	415(423)	2161	495	26(40)
13	500	50	20	500*500	35	15	3000	250(250)	6167	500	25(40)
14	1000	100	20	500*500	35	15	1000	500(500)	9080	1000	50(40)
11	500	10	10	100*100	25	5	500	500(500)	1697	500	50(60)
12	500	50	20	500*500	30	15	1000	425(441)	2131	495	27(60)
13	500	50	20	500*500	35	15	3000	250(250)	5962	500	25(60)
14	1000	100	20	500*500	35	15	1000	500(500)	8569	1000	50(60)

Table 4. Cost function of Experiment (11-14) using Swap & Reallocate move operators only in SA Algorithm.

EXP	#House	#ITAP-Location	ITAP-Cap	Grid-Area	WRC	W-Link	#Iterations	Cost-min(Avg)	RT/sec	Satisfied-House	#ITAP End(start)
11	500	10	10	100*100	25	5	500	500(500)	1811	500	50(50)
12	500	100	20	500*500	30	15	1000	748(1039)	1391	492	50(50)
13	500	50	20	500*500	35	15	3000	500(500)	6168	500	50(50)
14	1000	100	20	500*500	35	15	1000	500(500)	6203	1000	50(50)

Table 5. Cost function of Experiment (11-14) using all move operators with different number of ITAPs in HC Algorithm.

EXP	#House	#ITAP-Location	ITAP-Cap	Grid-Area	WRC	W-Link	#Iterations	Cost-min(Avg)	RT/sec	Satisfied-House	#ITAP End(start)
11	500	10	10	100*100	25	5	500	500(500)	175	500	50(50)
12	500	100	20	500*500	30	15	1000	635(667)	1545	495	48(50)
13	500	50	20	500*500	35	15	3000	470(496)	741	500	47(50)
14	1000	100	20	500*500	35	15	1000	500(500)	2152	1000	50(50)
11	500	10	10	100*100	25	5	500	3200(3380)	153	410	41(40)
12	500	100	20	500*500	30	15	1000	545(733)	1421	495	39(40)
13	500	50	20	500*500	35	15	3000	370(396)	2747	500	37(40)
14	1000	100	20	500*500	35	15	1000	5380(6234)	3723	840	42(40)
11	500	10	10	100*100	25	5	500	590(596)	237	500	59(60)
12	500	50	20	500*500	30	15	1000	735(755)	1555	495	58(60)
13	500	50	20	500*500	35	15	3000	580(594)	1050	500	58(60)
14	1000	100	20	500*500	35	15	1000	600(600)	5320	1000	60(60)

Table 6. Cost function of Experiment (11-14) using Swap & Reallocate move operators only in HC Algorithm.

EXP	#House	#ITAP-Location	ITAP-Cap	Grid-Area	WRC	W-Link	#Iterations	Cost-min(Avg)	RT/sec	Satisfied-House	#ITAP End(start)
11	500	10	10	100*100	25	5	500	500(500)	291	500	50(50)
12	500	100	20	500*500	30	15	1000	655(661)	2050	495	50(50)
13	500	50	20	500*500	35	15	3000	500(500)	2644	500	50(50)
14	1000	100	20	500*500	35	15	1000	500(500)	8518	1000	50(50)

7 Conclusion

This paper has presented a new two phase approach for solving the network optimization problem, and new heuristic move operators. We have demonstrated that the approach is highly successful for optimizing ITAP placements in WMNs. We verify from our result that our reallocate move always showed a better performance than the swap move, but it is slower. Our results show that our SA produces better results than our HC, but it takes slightly longer to run. Therefore, our implementations of SA algorithms produce near optimum solutions. The experimental evaluation shows the efficiency of a combination of all four move operators and it provides a better solution for the placement of ITAPs in WMNs. We will next look into bandwidth allocation for the house demand in order to achieve a good trade-off between fairness and throughput.

These abbreviations are used in the above tables;
EXP= Experiment
ITAP-Cap= ITAP Capacity/ Mbps
WRC= Wireless Range Connectivity/ M
W-Link= Wireless Link‘s Capacity/ Mbps
Cost= Cost Function
RT= Running Time/ Seconds
Real= Reallocate Move

References

1. Aoun, B., Boutaba, R., Member, S., Iraqi, Y., Kenward, G.: Gateway Placement Optimization in Wireless Mesh Networks With QoS Constraints. IEEE J. Sel. Areas Commun. 24(11), 2127–2136 (2006)
2. Wang, J., Xie, B., Cai, K., Agrawal, D.P.: Efficient mesh router placement in wireless mesh networks. In: 2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, pp. 1–9 (2007)
3. Chandra, R., Qiu, L., Jain, K., Mahdian, M.: Optimizing the placement of internet TAPs in wireless neighborhood networks. In: Proceedings of the 12th IEEE International Conference on Network Protocols, ICNP 2004, pp. 271–282 (2004)
4. Martignon, F., Paris, S., Filippini, I., Capone, A.: Efficient Bandwidth Allocation in Wireless Community Networks (2011)
5. Amaldi, E., Capone, A., Cesana, M., Filippini, I., Malucelli, F.: Optimization models and methods for planning wireless mesh networks. Comput. Networks 52(11), 2159–2171 (2008)
6. Allen, S.M., Cooper, I.M., Whitaker, R.M.: Optimising multi-rate link scheduling for wireless mesh networks. Comput. Commun. 35(16), 2014–2024 (2012)
7. Jun, J., Sichitu, M.L.: The Nominal Capacity of Wireless Mesh Networks. IEEE Wirel. Commun. 10, 8–14 (2003)
8. Capone, A., Filippini, I., Martignon, F.: Joint routing and scheduling optimization in wireless mesh networks with directional antennas. In: 2008 IEEE International Conference on Communications, pp. 2951–2957 (2008)
9. Benyamina, D., Ha, A., Gendreau, M.: Wireless Mesh Networks Design — A Survey. IEEE Communications Surveys & Tutorials 14(2), 299–310 (2012)

10. Xhafa, F., Sanchez, C., Barolli, L.: Ad hoc and neighborhood search methods for placement of mesh routers in wireless mesh networks. In: 2009 29th IEEE Int. Conf. Distrib. Comput. Syst. Work., pp. 400–405, June 2009
11. Fleszar, K., Osman, I.H., Hindi, K.S.: A variable neighbourhood search algorithm for the open vehicle routing problem. Eur. J. Oper. Res. 195(3), 803–809 (2009)
12. Xhafa, F., Sanchez, C., Barolli, L., Miho, R.: An annealing approach to router nodes placement problem in wireless mesh networks. In: 2010 Int. Conf. Complex, Intell. Softw. Intensive Syst, pp. 245–252, February 2010
13. Li, F., Wang, Y., Li, X.-Y., Nusairat, A., Wu, Y.: Gateway Placement for Throughput Optimization in Wireless Mesh Networks. Mob. Networks Appl. 13(1–2), 198–211 (2008)
14. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms, 2nd edn., vol. 7 (2001)
15. Rosenbluth, M.N., Teller, A.H., Metropolis, N., Rosenbluth, A.W.: Equation of state Calculations by Fast Computing Machines. J. Chem. Physic 21(6) (1953)
16. Manuscript, A.: NIH Public Access 21(suppl. 1), 1–19 (2006)