IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 12, 2016, accepted May 24, 2016, date of publication June 15, 2016, date of current version October 6, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2581187

Multi-Objective Topology Planning for Microwave-Based Wireless Backhaul Networks

YONGCHENG LI[1], ANLIANG CAI[2], GUANGYI QIAO[3], LEI SHI[3], SANJAY KUMAR BOSE[4], (Senior Member, IEEE), AND GANGXIANG SHEN[1], (Senior Member, IEEE)

[1] School of Electronic and Information Engineering, Soochow University, Suzhou 215006, China
[2] Department of Electronic Engineering, City University of Hong Kong, Hong Kong
[3] Huawei Technologies Company, Ltd., Shenzhen 518129, China
[4] Department of Electrical and Electronics Engineering, IIT Guwahati, Guwahati 781039, India

Corresponding author: G. Shen (shengx@suda.edu.cn)

ABSTRACT Wireless backhaul networks provide vital infrastructure support for large public and private networks, and their efficient design is essential for smoothly handling the rapid growth of the Internet traffic. For its efficient capacity utilization, a well-planned topology would be crucial for such networks. Existing studies on planning for wireless backhaul network topologies have mostly focused on a single parameter or on a few performance aspects, e.g., to minimize the network cost or to maximize the network reliability. It would, however, be more realistic to consider multiple performance aspects jointly, subject to a variety of system constraints for a microwave-based wireless backhaul network, as proposed in this paper. For the optimization, we formulate a general cost that incorporates the various performance aspects considered, based on different weight factors. We develop an integer linear programming (ILP) optimization model and also propose an efficient heuristic algorithm to plan cost-minimized tree topologies for both single stage and multi-stage design scenarios. This paper shows that the proposed heuristic algorithm is efficient to optimize multiple system objectives jointly and performs close to the ILP model. The performances of the topologies planned with the periodic constraint and the single stage scenario are also close to each other further confirming the efficiency of the proposed heuristic algorithm.

INDEX TERMS Network planning, wireless backhaul network, network optimization, periodic network planning.

I. INTRODUCTION

Mobile data traffic is widely expected to increase significantly with the vast deployment of the fourth generation (4G) mobile systems. To sustain this rapid growth, we need the wireless backhaul network to provide high capacity and high transmission efficiency. Fig. 1 depicts an example of a wireless backhaul network consisting of two parts, namely the core network and the access network [1]. The core network is made up of high-capacity switching equipment interconnected by a mesh of high-bandwidth links. The other part is the access network that carries traffic between user terminals and the core network nodes. In this study, we focus on the access network which supports the data transmission between many Base Transceiver Stations (BTSs) and a Base Station Controller (BSC). Each BTS forwards packets between its local users and the network, while the BSC is responsible for controlling all the BTSs and relaying the traffic from the BTSs to the core network.

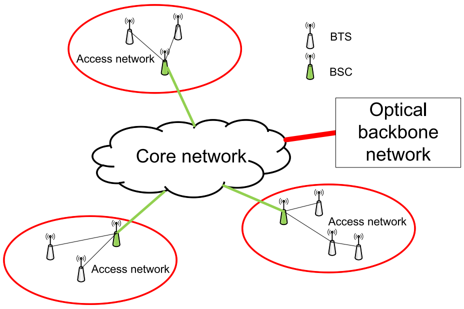


FIGURE 1. **An example of wireless backhaul network.**

Popular techniques currently in use for the wireless backhaul network include microwave- and fiber-based access systems. Here the wireless backhaul network refers to the X2 interface, i.e., the links between the base-stations, or between the base-stations and the base-station controllers.

Fiber-based access systems can offer high capacity over long distances. However, they are more expensive and take longer to deploy. In contrast, microwave-based systems are cheaper and can be implemented faster and more economically. Despite their lower capacity compared to fiber-based systems, microwave-based systems are in use for about 50% of all the wireless backhaul networks currently deployed [2]. A wireless backhaul network that connects multiple BTSs to a BSC is often deployed as a ring or a tree topology [3]. Due to their simplicity and cost-effectiveness, we focus on planning a *tree topology* for microwave-based wireless backhaul networks in this paper.

There have been many studies focusing on planning the topology for a wireless backhaul network [3]–[17]. These have generally targeted optimizing a single performance objective subject to a few system constraints so that the associated optimization problems are relatively straightforward and easy to tackle. However, a more realistic model for a wireless backhaul system would need to consider multiple performance aspects, subject to various system constraints. The topology optimization would need to consider multiple target objectives such as throughput, network efficiency, and deployment cost. The optimization constraints would also be diverse and would include both various hardware limitations and the bandwidth requirements of users. A more comprehensive study is, therefore, necessary to jointly take into account these multiple performance aspects and system constraints when planning a wireless backhaul topology. For this purpose, we develop a framework to model the wireless backhaul topology planning problem incorporating these multiple system performance aspects and constraints. In this paper, an integer linear programming (ILP) model and an efficient heuristic algorithm are developed to optimally plan a topology for the wireless backhaul network. The multiple optimization aspects jointly considered here include minimizations of the total numbers of long links,[1] traffic hops, link crosses along with small angles between neighboring links, and the sum of all the link distances. The constraints mainly include maximal nodal degree, maximal link level, maximal link distance, and a maximal number of nodes in each tree branch. The detailed definitions for these performance aspects and constraints are elaborated on in a later section. Moreover, to consider the issue involved in the growth of a wireless backhaul network, we also propose a *multi-stage (or periodic) constraint*, under which a node in an earlier planning stage cannot be connected as a child to a node of a later stage. Simulation studies over a wide range of test networks demonstrate that the proposed heuristic algorithm is efficient and performs close to the optimum ILP model for both cases with and without the multi-stage constraint.

The remainder of this paper is organized as follows. In Section II, we survey the existing literature on the related works. In Section III, we define the problem of wireless backhaul topology planning and formulate a ''general cost'' to balance multiple optimization aspects. In Section IV, we present the ILP models for single-stage (non-periodic) and multi-stage (periodic) wireless backhaul network planning. In Section V, we propose an efficient heuristic algorithm for wireless backhaul planning which consists of two steps, i.e., finding an initial solution and subsequent re-optimization process. In Section VI, we present and discuss the simulation results obtained. We conclude the paper in Section VII.

II. LITERATURE REVIEW

Considerable efforts have already been made on optimal topology planning for microwave-based wireless backhaul networks. To plan the access network, the first step is to determine the required number of BSCs and their locations, which can be classified as a facility location and clustering problem. Harmatos *et al.* [5] proposed an algorithmic method to optimize the number and the locations of BSCs jointly with the network topology. Based on this, Jüttner *et al.* [6] proposed an improved algorithm to determine the cost-optimal number and locations of BSCs which achieves a better performance. Wu and Pierre [7], [8] also proposed an efficient constraint-based optimization model to find the optimal locations of BSCs. Lauther *et al.* [9] presented two new clustering approaches based on a proximity graph to partition a set of given BTSs into a near optimum number of BSC-clusters.

The second step is to find an optimal BTSs network topology for each BSC node, i.e., single tree topology planning. For this, several studies take the network reliability and the deployment cost of the wireless backhaul network into account. Tipper *et al.* [10], [11] discussed the effects of reliability issues and presented a novel network design model that incorporates the influence of user mobility for a fault-tolerant wireless backhaul network. Dengiz *et al.* [12] presented a generic algorithm to optimize the design of reliable networks. Gódor *et al.* [3] considered the cost-optimal topology planning problem and proposed a heuristic algorithm relying on iterative problem decomposition, clustering methods, and local optimization. In [13], they also proposed a heuristic algorithm based on a combination of an adaptive version of the Simulated Annealing meta-heuristic and a local improvement strategy to plan a multi-constrained and capacitated sub-network tree.

Some earlier works have also focused on the tradeoff between low cost and network reliability. Jan [14] proposed heuristic algorithms to minimize the network cost with a constraint of guaranteeing reliable communications of all terminals. Since these optimization algorithms are computationally infeasible for large networks, Szlovencsak *et al.* [1] proposed two types of heuristic algorithms to minimize the cost of the wireless backhaul network while guaranteeing a specific reliability level. Nadiv and Tzvika [15] took a close look at the pros and cons of tree and ring topologies for a wireless backhaul network, with particular attention to the associated cost consideration. Kuo *et al.* [16] compared different wireless backhaul network topology options,

[1] Here ''link'' refers to a ''microwave link.''

i.e., mesh, ring, and tree, with regard to network performance and deployment cost.

St-Hilaire [17] made a comprehensive literature survey on the topological planning problem for the wireless backhaul network. Recent research has also examined heterogeneous network or fiber-wireless (FiWi) access networks which combine both microwave and fiber links [3], [18], [19]. Other associated works for planning microwave wireless backhaul networks may also be found in [20]–[22].

III. PROBLEM OF WIRELESS BACKHAUL NETWORK PLANNING

A. NETWORK MODEL

This paper focuses on single tree topology planning for the wireless backhaul network, i.e., designing a BTS tree for a single BSC node. As given parameters, there is one root node, i.e., BSC, and multiple non-root nodes, i.e., BTSs. The geographical position of each node is also given, based on which visibility between each pair of nodes can be decided. Here a pair of nodes is visible if a microwave link can be established between the nodes. This visibility is subject to the distance between the two nodes and the topography of the area that they are located. If the two nodes are too far away from each other or if there is a hill, mountain, or tall building between them, then they will also not be visible to each other.

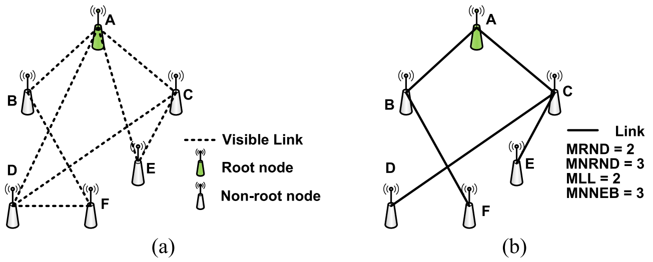


FIGURE 2. An example of wireless backhaul topology planning. (a) An initial mesh topology. (b) A planned tree topology.

Fig. 2(a) shows a six-node wireless backhaul network example which consists of one root node A (BSC) and five non-root nodes B, C, D, E, and F (BTSs). If any two nodes are visible to each other, then there is a link represented by a dotted line connecting them. In this example, there are eight connected links. Based on this initial mesh topology, the problem of optimal tree planning is to find an optimal tree topology that is rooted at the root node and connects all the non-root nodes. As an example, Fig. 2(b) shows a planned tree topology for this network.

The tree planning problem has different optimization objectives and is subject to a variety of system constraints which are introduced as follows.

B. OBJECTIVES

1) OBJECTIVE 1 (LINK CROSS)

If two microwave links between various node pairs cross each other, we call the situation *link cross*. Fig. 3(a) shows an example of link cross between links (B-F) and (C-D). The physical significance of link cross corresponds to the inefficiency of spectrum resource utilization. Since a topology with fewer link crosses would generally correspond to more efficient spectrum resource utilization, we would prefer a topology with fewer link crosses.

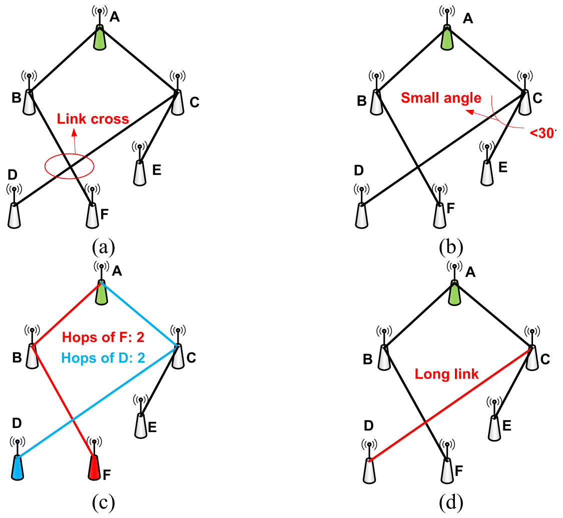


FIGURE 3. Examples of different performance aspects. (a) Link cross. (b) Small angle. (c) Traffic hop. (d) Long link.

2) OBJECTIVE 2 (SMALL ANGLE)

Neighboring microwave links incident to a common node can interfere with each other if their formed angle is too small. In that case, we would require the assignment of different frequencies to each of the links which would, therefore, consume more frequency resources. It would therefore be important to minimize the total number of small angles formed by neighboring links incident to a common node. In this study, we assume that any angle smaller than 30 degrees is a small one. Fig. 3(b) shows an example of a small angle ∠D-C-E that is formed by links (C-D) and (C-E).

Note that in this study, we assume that the designed topology needs to be general to support any type of access technique. A time division-based access technique may ignore this constraint by properly coordinating the ON-OFF transmission of two neighboring links with a small angle. This would, however, significantly increase the control complexity and may also impact the overall spectrum utilization of the whole system.

3) OBJECTIVE 3 (TRAFFIC HOP)

End-to-end delay is an important performance criterion for user applications, which can also indirectly reflect on the throughput of the overall wireless backhaul system. Traffic hops from a root node to a non-root node can be used to measure the end-to-end delay and the system throughput. In Fig. 3(c), the numbers of hops between the root node and the non-root nodes D and F are both two. To minimize the end-to-end delay and maximize the system throughput, as an important performance objective, we need to minimize the average (or total) number of traffic hops between the root node and the non-root nodes in a planned tree.

4) OBJECTIVE 4 (LONG LINK)

To establish a long microwave link, a large antenna and a high transmission power are required, which leads to a high system cost. To minimize the overall system cost, it is vital to minimize the total number of long links in the system. In this study, we consider a link as a long link for the following three situations: (1) A link longer than 20 km is considered as a long link. (2) A link which is among the longest 20% incident or outbound links of a node is considered as a long link. (3) Following (2), among the remaining 80% incident or outbound links of the node, a link that is longer than the average distance of these links is considered as a long link. In Fig. 2(a), link (C-D) is considered as a long link because it satisfies the conditions (2) and (3).

5) OBJECTIVE 5 (LINK DISTANCE)

To minimize the total cost of the whole system, it is important to minimize the sum of link distances in a planned tree.

In general, it is relatively easy to minimize one of these objectives as was done in the earlier studies of [3] and [12]. However, it is challenging to optimize all the above objectives jointly due to the difficulty in properly balancing all the performance aspects. For this, we define a "general cost" for a planned tree topology, given by (1)

$$Cost = w_h \cdot N_h + w_d \cdot D + w_l \cdot N_l + w_s \cdot N_s + w_c \cdot N_c \quad (1)$$

where N_h is the sum of the traffic hops from each non-root node to the root node, D is the sum of the link distances, and N_l, N_s, and N_c are the numbers of long links, small angles, and link crosses in the tree topology, respectively. In addition, w_h, w_d, w_l, w_s, and w_c are the respective weight factors of each performance aspect. Obviously, it is important to properly choose a value for each of them in order to well balance jointly all the performance aspects.

Note that for the two constraints on long link and link distance, this study focuses only on the line-of-sight (LOS) case. For better spectrum efficiency, it may also be interesting to consider the non-line-of-sight (NLOS) case, in which a spectral efficiency based optimization would be more practical than a distance based one. The NLOS case has not been considered by us in this paper but will be considered in our subsequent research work.

C. CONSTRAINTS

The wireless backhaul network is subject to various system constraints. First, each node in a wireless backhaul network may only have a limited number of nodes connected to it, which is called its *maximal nodal degree*. This may be constrained by the number of switch ports in hardware. The degree constraint for a root node is referred to as the *Maximal Root Nodal Degree* (*MRND*) while for a non-root node, it is called the *Maximal Non-Root Nodal Degree*, (*MNRND*). In Fig. 2(b), the root node connects up to 2 nodes, i.e., $MRND = 2$ while the non-root nodes connects up to 3 nodes, i.e., $MNRND = 3$.

Second, to meet the requirements of a maximal end-to-end delay and a minimum system throughput, there can be a constraint called *Maximal Link Level* (*MLL*), which is defined as the maximal number of traffic hops from a root node to a non-root node. The tree in Fig. 2(b) has $MLL = 2$.

Third, to balance the traffic load in the wireless backhaul system, there can be an upper bound on the number of non-root nodes connected to a root node through each of its branches, called the *Maximal Number of Nodes in Each Branch* (*MNNEB*). In Fig. 2(b), the tree has $MNNEB = 3$ on the right-hand branch.

There can also be other constraints such as the limited capacity of each microwave link and survivability requirements. However, this study has only considered the above three key constraints. Other constraints can be similarly incorporated, as required.

D. MULTI-STAGE PLANNING

A wireless backhaul network can be deployed in multiple stages. A group of non-root nodes closer to the root node may be deployed first while the other non-root nodes may be incrementally deployed in the future. In this context, a node of an earlier stage should not be connected (as a child node) to a node of a later stage. It is also important to plan a tree topology that can incorporate such a multi-stage implementation. We refer to this type of tree planning as *multi-stage (or periodic) planning*. Fig. 4 shows an example of multi-stage planning, in which all the non-root nodes are divided into three stages, i.e., from stages 1 to 3. Because node B belongs to the second stage, it cannot be connected to node D of the third stage. Rather, it should be attached to either node A or C of the first and second stages.

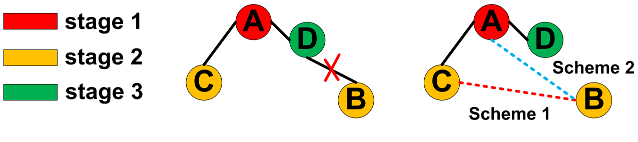


FIGURE 4. An example of the multi-stage constraint.

IV. OPTIMAL PLANNING FOR THE WIRELESS BACKHAUL TOPOLOGY

In this section, we first present the research problem of wireless backhaul topology planning, which is followed by an ILP model for the solution to the problem.

A. PROBLEM STATEMENT

We plan a topology for the wireless backhaul network aiming to jointly optimize multiple performance aspects subject to various system constraints. The given inputs of the problem are as follows:

1) The physical topology of a wireless backhaul network $\boldsymbol{G} = (N, L)$, where $\boldsymbol{N}$ is the set of network nodes and $\boldsymbol{L}$ is the set of network links. The root node is denoted as r and all the non-nodes are included in a set $\boldsymbol{M}$.
2) Based on the geographical location of each node, we can find the distance of each link between nodes c and k as $DS_{c,k}$. We also indicate whether two links $(c1, k1)$ and $(c2, k2)$ cross each other using a binary variable $ITS_{c2,k2}^{c1,k1}$. Specifically, if the two links do cross each other, then $ITS_{c2,k2}^{c1,k1}$ equals 1; otherwise, it is 0. Similarly, we indicate whether two links form a small angle using the binary variable $AGL_{c2,k2}^{c1,k1}$. Specifically, if the two links do form a small angle, then $AGL_{c2,k2}^{c1,k1}$ equals 1; otherwise, it is 0.

The constraints of the optimization problem are as follows: (1) The maximal nodal degrees of the root node and non-root nodes, i.e., *MRND* and *MNRND*; (2) The maximal link level, *MLL*; (3) The maximal number of nodes in each branch, *MNNEB*.

The outputs of the optimization problem include an optimal tree topology that connects all the non-root nodes to the root node. The tree has a minimal "general cost" given by (1), which jointly (in a balanced way) optimizes the total numbers of traffic hops, long links, link crosses, small angles, and the sum of the link distances.

B. ILP MODEL FOR SINGLE STAGE PLANNING

In this section, we present the ILP model for single stage topology planning of a wireless backhaul network. The sets, parameters, and variables are as follows.

Sets:

N The set of network nodes in a wireless backhaul network, including the root node and the non-root nodes.

M The set of non-root nodes in the network.

L The set of links in the wireless backhaul network. Note that between each visible node pair, there is a pair of bi-directional links.

Parameters:

MRND The maximal nodal degree of the root node.

MNRND The maximal nodal degree of each non-root node.

MLL The maximal number of hops from the root node to each non-root node.

MNNEB The maximal number of non-root nodes in each branch of a tree topology.

r The root node of the network.

$DS_{c,k}$ Link distance between nodes c and k, $\forall c, k \in N$.

$ITS_{c2,k2}^{c1,k1}$ Link cross parameter that takes the value of 1 if links $(c1, k1)$ and $(c2, k2)$ crosses each other; 0, otherwise. If the two links share any common node, their link cross parameter is 0.

$AGL_{c2,k2}^{c1,k1}$ Small angle parameter that takes the value of 1 if the two neighboring links $(c1, k1)$ and $(c2, k2)$ form a small angle; 0, otherwise. Here these two links must share one and only one common node.

$LG_{c,k}$ Long link parameter that takes the value of 1 if a link is a long one; 0, otherwise.

Δ A big number.

Variables:

$P_{n,r,c,k}$ A binary variable that equals 1 if link (c, k) is traversed by the path from a non-root node n $(n \in M)$ to the root node r; 0, otherwise.

$SLT_{c,k}$ A binary variable that equals 1 if link (c, k) is included in the planned tree topology; 0, otherwise.

$X_{c2,k2}^{c1,k1}$ A binary variable that equals 1 if links $(c1, k1)$ and $(c2, k2)$ cross each other and are both included in the tree topology; 0, otherwise. Here these two links cannot share any common node.

$SA_{c2,k2}^{c1,k1}$ A binary variable that equals 1 if links $(c1, k1)$ and $(c2, k2)$ form a small angle and are both included in the tree topology; 0, otherwise. Here these two links must share one and only one common node.

Objective: Minimize

$$w_h \cdot \sum\nolimits_{n\in \boldsymbol{M},(c,k)\in \boldsymbol{L}} P_{n,r,c,k} + w_d \cdot \sum\nolimits_{(c,k)\in \boldsymbol{L}} (SLT_{c,k} \cdot DS_{c,k}) + w_l \cdot \sum\nolimits_{(c,k)\in \boldsymbol{L}} (SLT_{c,k} \cdot LG_{c,k}) + w_s \cdot \sum\nolimits_{(c1,k1),(c2,k2)\in \boldsymbol{L},|\{c1,k1\}\cap\{c2,k2\}|=1} SA_{c2,k2}^{c1,k1}/2 + w_c \cdot \sum\nolimits_{(c1,k1),(c2,k2)\in \boldsymbol{L},|\{c1,k1\}\cap\{c2,k2\}|=0} X_{c2,k2}^{c1,k1}/2 \tag{2}$$

Constraints:

$$\sum\nolimits_{(c,k)\in \boldsymbol{L}} P_{n,r,c,k} = \begin{cases} 1 & c = n \text{ or } k = r \\ 0 & c = r \text{ or } k = n, \end{cases} \quad \forall n \in \boldsymbol{M} \tag{3}$$

$$\sum\nolimits_{(c,x)\in \boldsymbol{L}} P_{n,r,c,x} = \sum\nolimits_{(x,k)\in \boldsymbol{L}} P_{n,r,x,k}, \quad \forall n,\ x \in \boldsymbol{M},\ x \neq n \tag{4}$$

$$\Delta \cdot SLT_{c,k} \geq \sum\nolimits_{n\in \boldsymbol{M}} P_{n,r,c,k}, \quad \forall (c,k) \in \boldsymbol{L} \tag{5}$$

$$SLT_{c,k} + SLT_{k,c} \leq 1, \quad \forall (c,k), (k,c) \in \boldsymbol{L} \tag{6}$$

$$\sum\nolimits_{(c,k)\in \boldsymbol{L}} SLT_{c,k} = 1, \quad \forall c \in \boldsymbol{M} \tag{7}$$

$$\sum\nolimits_{(c,k)\in \boldsymbol{L}} SLT_{c,k} = |\boldsymbol{N}| - 1 \tag{8}$$

$$\sum\nolimits_{(c,k)\in \boldsymbol{L}} P_{n,r,c,k} \leq MLL, \quad \forall n \in \boldsymbol{M} \tag{9}$$

$$\sum\nolimits_{(c,r)\in \boldsymbol{L}} SLT_{c,r} \le MRND \tag{10}$$

$$\sum\nolimits_{(c,k)\in \boldsymbol{L}} SLT_{c,k} \le MNRND - 1 \quad \forall k \in \boldsymbol{M} \tag{11}$$

$$\sum\nolimits_{n\in \boldsymbol{M}} P_{n,r,c,r} \le MNNEB, \quad \forall (c,r) \in \boldsymbol{L} \tag{12}$$

$$\begin{aligned} X^{c1,k1}_{c2,k2} \ge ITS^{c1,k1}_{c2,k2} &+ SLT_{c1,k1} + SLT_{c2,k2} - 2, \\ &\forall\left(c1,k1\right), \left(c2,k2\right) \in \boldsymbol{L}, \\ &\left|\left\{c1,k1\right\} \cap \left\{c2,k2\right\}\right| = 0 \end{aligned} \tag{13}$$

$$\begin{aligned} SA^{c1,k1}_{c2,k2} \ge AGL^{c1,k1}_{c2,k2} &+ SLT_{c1,k1} + SLT_{c2,k2} - 2, \\ &\forall\left(c1,k1\right), \left(c2,k2\right) \in \boldsymbol{L}, \\ &|\{c1,k1\} \cap \{c2,k2\}| = 1 \end{aligned} \tag{14}$$

In (1), w_h, w_d, w_l, w_s, and w_c are the respective weight factors of all the performance aspects. The value of each factor can be defined according to the importance of each aspect. In addition, when links $(c1, k1)$ and $(c2, k2)$ form a small angle, $SA^{c1,k1}_{c2,k2} = 1$ and $SA^{c2,k2}_{c1,k1} = 1$. Thus, when finding the total number of small angles, we need to divide $\sum_{(c1,k1),(c2,k2)\in L,|\{c1,k1\}\cap\{c2,k2\}|=1} SA^{c1,k1}_{c2,k2}$ by 2 as $\sum_{(c1,k1),(c2,k2)\in L,|\{c1,k1\}\cap\{c2,k2\}|=1} SA^{c1,k1}_{c2,k2}/2$. For the same reason, $\sum_{(c1,k1),(c2,k2)\in L,|\{c1,k1\}\cap\{c2,k2\}|=0} X^{c1,k1}_{c2,k2}/2$ is defined as the total number of link crosses, in which there is also a denominator 2. Here $|\{c1, k1\} \cap \{c2, k2\}| = 1$ means that links $(c1, k1)$ and $(c2, k2)$ share only one common node, while $|\,\{c1, k1\} \cap \{c2, k2\}\,| = 0$ means that the two links do not share any common node.

Constraint (3) ensures that there is one egress link from a starting non-root node n and one ingress link at an ending root node r of a path from node n to r. Constraint (4) ensures that the number of ingress links of an intermediate node equals the number of its egress links for all the paths between any source-destination (SD) node pair from n to r. Constraint (5) says that link (c, k) is included in the final planned tree topology if it is traversed by the path between the root and a non-root node. Constraint (6) ensures that each link in the planned tree topology is unidirectional towards the root node. Constraint (7) ensures that every non-root node has only one parent node. Constraint (8) says that the total number of links in a tree topology is equal to the total number of nodes in the tree topology minus one. Constraint (9) limits the link level for each non-root node, i.e., a limited number of hops from the root node to each non-root node. Constraints (10) and (11) ensure the limits of maximal nodal degrees for the root and non-root nodes. Constraint (12) ensures the limit on the total number of non-root nodes in each tree branch of the root node. Constraints (13) and (14) decide whether links $(c1, k1)$ and $(c2, k2)$ form a link cross or a small angle, respectively.

The computational complexity of the ILP model is decided by the dominant numbers of variables and constraints. The ILP model has a total of $O\left(|L|^2\right)$ variables and $O\left(|L|^2\right)$ constraints, where $|L|$ is the total number of unidirectional links in the network.

C. ILP MODEL FOR MULTI-STAGE PLANNING

For multi-stage topology planning, we need to consider the constraint of node stage when connecting non-root nodes to a tree topology. More specifically, a node of an earlier stage should not be connected as a child to a node of a later stage. Taking into account the multi-stage constraint, we have an extended ILP model as follows.

In addition to the sets, parameters, and variables of the single stage model, we define a new parameter S_n to denote the stage of each node. Also, we need a new constraint that ensures the stage relationship between the nodes in a tree topology. The parameter and constraint are as follows.

Extra parameter:

S_n The stage of node n, $\forall n \in N$;

Extra constraint:

$$S_c \ge S_k \cdot SLT_{c,k}, \quad \forall\left(c,k\right) \in L \tag{15}$$

The ILP model for multi-stage planning has the same objective as that of the single stage one. The two models also have the same computational complexity.

V. HEURISTIC ALGORITHM FOR WIRELESS BACKHAUL TOPOLOGY PLANNING

Though the ILP models can provide optimal solutions to the planning problem, it would have high computational complexity and, therefore it would be difficult to find optimal solutions for large networks using this optimum approach. Thus, it is necessary to develop an efficient heuristic algorithm for multi-objective topology planning. In this section, we propose a heuristic algorithm which mainly consists of two steps, i.e., finding an initial solution (i.e., Step A) and re-optimizing the initial solution (i.e., Step B). The two steps are given next.

A. FINDING AN INITIAL SOLUTION

In this step, we employ a modified Dijkstra's shortest path searching algorithm to find the shortest route from the root to each non-root node subject to the set of constraints given in Section III.C. We merge all the links traversed by the route thus found to form a tree topology. We call the algorithm Dijkstra's Backhaul Topology Planning (DBTP) algorithm. The detailed steps of the algorithm are as follows:

In the above algorithm, when the searching process reaches an intermediate node i (however, before reaching destination node m), which forms a sub-route from source node r to i, denoted as R, for the subsequent searching process, we set the cost of each link l in the initial network topology $\boldsymbol{G}$ as follows. If the link has been included in the planned tree graph $\boldsymbol{G^T}$, we set its cost to be d_l which is the link distance. Otherwise, the link cost is calculated by (16).

$$\begin{aligned} Cost_l = w_h \cdot N_h + w_d \cdot d_l + w_l \cdot \sigma_l + w_s \\ \cdot \left(N^s_l + N^{s,R}_l\right) + w_c \cdot \left(N^c_l + N^{c,R}_l\right) \end{aligned} \tag{16}$$

Algorithm 1 Dijkstra's Backhaul Topology Planning (DBTP) Algorithm

```
Input and initialization:
      $\boldsymbol{G} = (\boldsymbol{N}, \boldsymbol{L})$   // network topology
      $\boldsymbol{M}$              // non-root node list
      $r$              // root node
Output:
      $\boldsymbol{G^T} = (\boldsymbol{N^T}, \boldsymbol{L^T})$   // a planned tree topology
  for each $m \in M$ do
      if $m! \in G^T$ then
          Use a modified Dijkstra's algorithm to
           find the shortest route $p(r, m)$ on $\boldsymbol{G}$ subject
           to the constraints described
           in Section III. C;
          if $p(r, m)$ exists then
              Add all $l \in p(r, m)$ to $\boldsymbol{G^T}$ subject to
               the constraint of maximal nodal
               degree;
               Update the cost of each link $l$ that is
               newly added to $\boldsymbol{G^T}$ to be $d_l$;
          else
              Add $m$ to an isolated node list; //the
               list records the nodes that cannot
               be included in $\boldsymbol{G^T}$
          end if
      end if
  end for
```

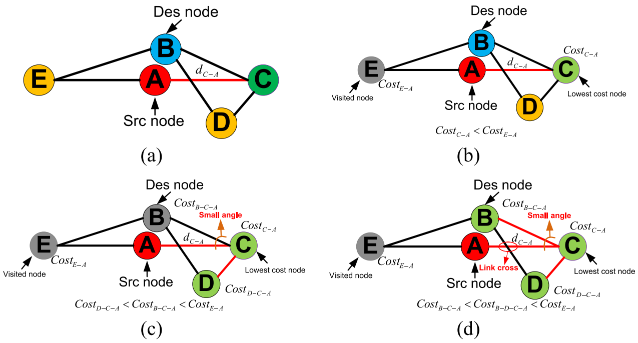


FIGURE 5. An example of the modified Dijkstra's algorithm. (a) An example of searching the route between nodes A and B. (b) Adding the lowest cost node C. (c) Adding the lowest cost node D. (d) Adding the lowest cost node B.

where each term has the following meaning:

N_h The hop count of sub-route R from root node r to current node i plus one; the usage of this term aims to minimize the link level of a tree topology.

d_l Physical distance of link l in units of km;

σ_l Long link status which equals 1 if link l is a long link; otherwise it is 0;

N_l^s Number of small angles formed between link l and all the links in $\boldsymbol{G^T}$;

$N_l^{s,R}$ Number of small angles formed between link l and all the links on route R but not in $\boldsymbol{G^T}$;

N_l^c Number of link crosses between link l and all the links in $\boldsymbol{G^T}$;

$N_l^{c,r}$ Number of link crosses between link l and all the links on route R but not in $\boldsymbol{G^T}$;

Based on the above link costs, we further iteratively take subsequent searching steps until we reach destination node m.

We use the example in Fig. 5 to illustrate the searching process for the modified Dijkstra's algorithm. Fig. 5(a) shows an initial network, in which node C has been added in $\boldsymbol{G^T}$ and thus its cost was updated to be d_{C-A}. In addition, in order to simplify the searching process, we assume that link (A-E) has a very long distance and therefore a very high cost.

To search for a lowest-cost path between source node A and destination node B, we first create an empty visited node list Θ. Starting from A, we scan its neighboring nodes to add C and E to Θ. Meanwhile, we calculate the cost for each node in $\Theta = \{C, E\}$ to choose C (see Fig. 5(b)) as the current lowest cost node from A (because of a very high cost of link (A-E)) and update the current sub-route R as A-C. We remove C from Θ, and then starting from C, we scan its neighboring nodes B and D and add them to Θ, which gives $\Theta = \{E, B, D\}$. The total cost from B to A via link (B-C) is calculated as $w_h \cdot 2 + w_d \cdot d_{B-C} + w_s + d_{C-A}$ since from B to A there are two hops and ∠B-C-A is a small angle. Similarly, the total cost from D to A via link (C-D) is calculated as $w_h \cdot 2 + w_d \cdot d_{D-C} + d_{C-A}$. Comparing the costs (to A) of all the nodes in Θ, we choose D as the current node (see Fig. 5(c)) due to its lowest cost and set the current sub-route R as A-C-D. We remove D from Θ, and then start from D, we have B as its neighboring node and calculate the cost from B to A via link (B-D) as $w_h \cdot 3 + w_d \cdot d_{B-D} + w_c + Cost_{C-D} + d_{C-A}$ since there is a link cross between (B-D) and (A-C). Because this new cost via link (B-D) is higher than the previous recorded one, i.e., $w_h \cdot 2 + w_d \cdot d_{B-C} + w_s + d_{C-A}$, we do not need to update the total cost to A for B. Then, comparing the costs (to A) of all the nodes in Θ, we choose B as the current node and decide that its lowest cost to A should be $w_h \cdot 2 + w_d \cdot d_{B-C} + w_s + d_{C-A}$ via link (B-C). Consequently, we find that the route with the lowest cost from A to B should be A-C-B (see Fig. 5(d)).

With the DBTP algorithm, we can obtain a tree topology $\boldsymbol{G^T}$. In addition, with different node sequences in the non-root node list $\boldsymbol{M}$, different tree topologies $\boldsymbol{G^T}$ can be found. The topologies thus found show different optimality in their performance aspects. This therefore requires us to run the DBTP algorithm multiple times for the same non-root node list with different node sequences so that we can then choose the one with the best performance. Fig. 6 illustrates the diagram of how to shuffle the node list and choose a tree graph with the lowest "general cost" according to (1). This tree becomes the final solution of Step A.

In general, if more shuffled node sequences are considered, more tree topologies can be found, and a better tree topology with a lower general cost can be expected with the above

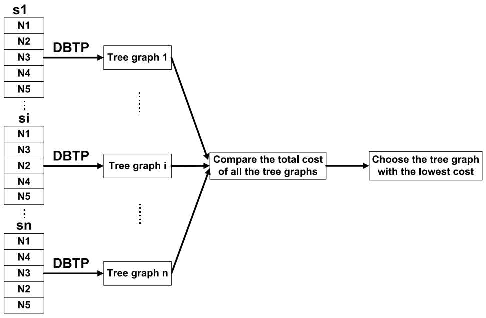

s1
N1
N2
N3
N4
N5
DBTP
Tree graph 1
si
N1
N3
N2
N4
N5
DBTP
Tree graph i
sn
N1
N4
N3
N2
N5
DBTP
Tree graph n
Compare the total cost of all the tree graphs
Choose the tree graph with the lowest cost

FIGURE 6. Diagram of multi-iteration solution selection process.

multi-iteration process. This will, of course, be at the expense of a longer overall computation time.

B. RE-OPTIMIZING THE INITIAL SOLUTION

The re-optimization process contains four steps to ensure a low "general cost" of the final tree topologies. These steps are as follows:

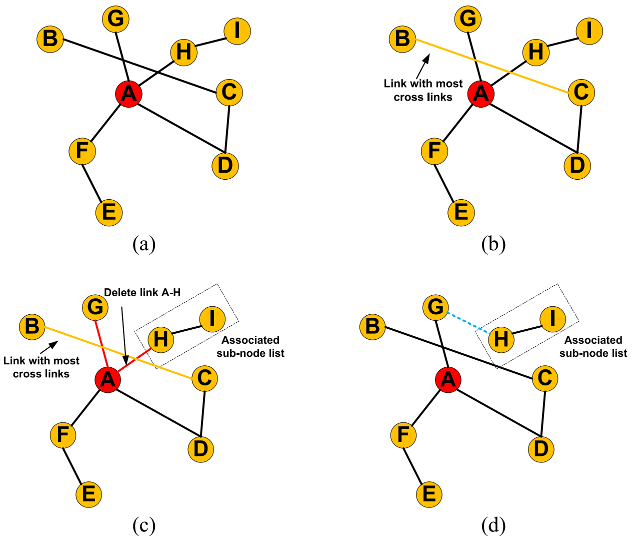


FIGURE 7. An example of link cross deletion. (a) An example of link cross. (b) Link with the largest number of link crosses. (c) Deleting link A-H. (d) Adding a new link G-H.

1) DELETE LINK CROSSES

This step is to reduce the total number of link crosses for an initial tree graph obtained by Step A. It consists of two sub-steps, i.e., deleting link crosses and reconnecting isolated sub-trees. Fig. 7 shows an example of link cross deletion. Fig. 7(a) shows an initial tree topology, in which there are two link crosses. To delete the link crosses, we first find the link that has the most crosses, i.e., link (B-C). We add all the links that crosses link (B-C) to a list S, in which links (A-H) and (A-G) are included. We retrieve link A-H from S and remove it from the tree graph as shown in Fig. 7(c). The removal of link A-H creates an isolated sub-tree (or sub-node list) ⊥(H-I) that contains nodes H and I. We need to reconnect the sub-tree back to the parent tree. Different new tree topologies can be constructed when ⊥(H-I) is reconnected through different links. We need to select a new tree topology with the lowest "general cost" as defined by (1). In Fig. 7(d), the new tree topology that reconnects node G to node H has the lowest "general cost." Hence, we reconnect the sub-tree back to the parent tree through node G. We can repeat the same process to remove link cross between (A-G) and (B-C) by reconnecting B to G.

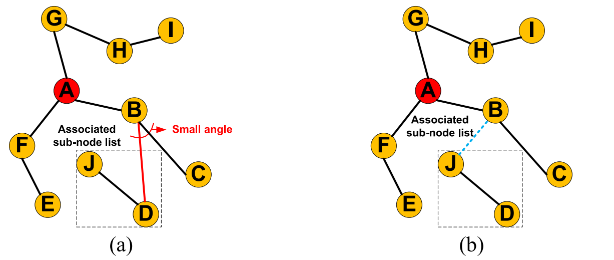


FIGURE 8. An example of small angle removal. (a) Deleting a longer link A-D. (b) Adding a new link A-C.

2) REMOVE SMALL ANGLES

In this step, for each small angle, we remove a longer link that forms the small angle and then reconnect the isolated sub-tree (due to the link removal) back to the parent tree. In Fig. 8(a), we remove the longer link (B-D) of a small angle ∠D-B-C, which leads to an isolated sub-tree ⊥(J-D). We then select a new link (B-J) to reconnect ⊥(J-D) back to the parent tree and construct a new tree topology which has a lower "general cost." The removal process is repeated until no small angles can be removed.

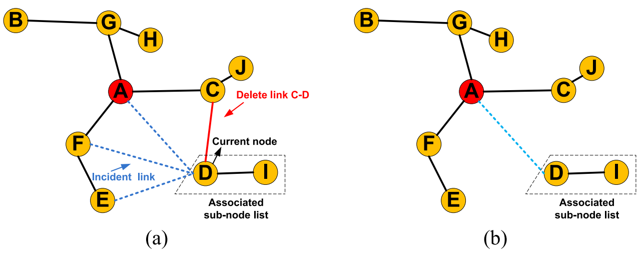


FIGURE 9. An example of reducing traffic hops. (a) Deleting link C-D. (b) Adding a new link A-D.

3) REDUCE TRAFFIC HOPS

This step is to reduce the total number of hops between each non-root node and the root node. Specifically, for each of the non-root nodes, we keep on modifying its connecting link to the parent tree to see if the total number of hops can be reduced. Fig. 9 shows an example of this step. For link (C-D), we set node D (a node farther away from the root node) as the current node. Then, from all the incident links of the current node, we find the one that can construct a tree topology with a smaller number of traffic hops. In this example, though link (C-D) is shorter than link (A-D) in distance, a tree topology containing link (A-D) has a smaller number of traffic hops than that containing link (C-D). Thus, we remove link (C-D) and reconnect the isolated sub-tree ⊥(D-I) back to the parent tree through link (A-D).

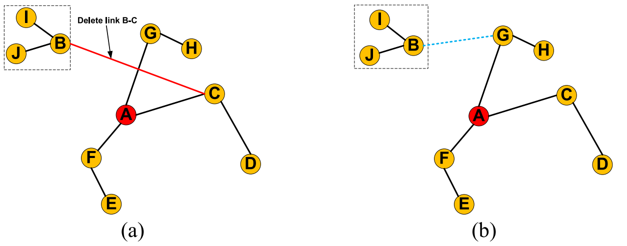


FIGURE 10. An example of reducing "general cost." (a) Deleting link B-C. (b) Adding a new link B-G.

4) REDUCE "GENERAL COST"

Unlike Step (3), this step keeps on reducing the "general cost" of a tree graph by repeatedly deleting a link and then reconnecting an isolated sub-tree back to the parent tree. Each reconnecting process leads to a different new tree topology. If any newly obtained tree topology has a lower "general cost" than that of an old one, we keep this new one and then continue the same deleting and reconnecting process. Fig. 10 shows such an example, in which link (B-C) is removed, and the isolated sub-tree ⊥(B-I-J) is reconnected back to the parent node through link (B-G) which leads to a new tree topology that has a lower "general cost" than that of the old tree. Thus, we have a new tree topology as shown in Fig. 10(b). This step is a final one to further tune down the "general cost" after we have optimized the topology based on each performance aspect.

In each of the re-optimization steps, if the "general cost" of a new tree topology after re-optimization is found to be higher than or equal to that of the tree topology before the re-optimization, we consider the re-optimization step invalid and restore the original tree topology. We also continue repeating the first two steps, i.e., deleting link crosses and removing small angles, until no link crosses or small angles can be reduced. After that, the subsequent two steps are repeated for a certain number of times (100 times in this study) due to the high computational time involved. In each case, all the links on a tree topology are scanned for improving the performance aspects, i.e., traffic hops and general cost.

Due to the extended Dijkstra's algorithm, the computational complexity of finding an initial solution is $O\left(|M|\,|L_{GT}|+|L|\right)$, where $|M|$ is the total number of non-root network nodes, $|L_{GT}|$ is the number of links in a planned tree graph, and $|L|$ is the total number of unidirectional links in the network. The re-optimization process subsequent to finding the initial solution has a computational complexity of $O\left(|L|^2\right)$.

C. MULTI-STAGE TOPOLOGY PLANNING

For the case of multi-stage topology planning, we need to add a multi-stage (periodic) constraint to ensure that a node of an earlier stage must not be connected (as a child node) to a node of a later stage. This constraint can be easily incorporated into the DBTP algorithm when searching for an initial tree topology. Specifically, a node of a later stage should not become the parent of a node of an earlier stage in the course of Dijkstra's algorithm based searching.

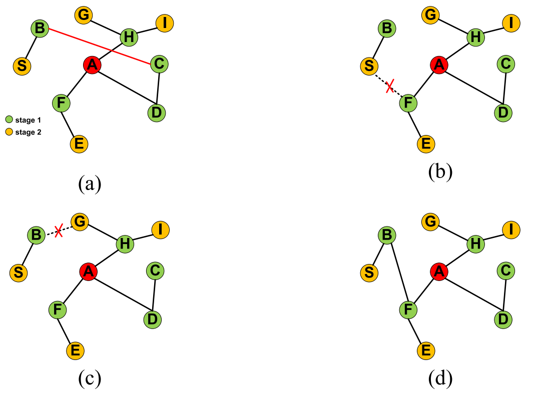


FIGURE 11. The periodic constraint in the re-optimization process. (a) An initial tree graph. (b) Connecting ⊥(B-S) back to the parent tree through (S-F) (*illegal*). (c) Connecting ⊥(B-S) back to the parent tree through (B-G) (*illegal*). (d) Connecting ⊥(B-S) back to the parent tree through (B-F) (*legal*).

In the re-optimization steps, we need to examine the periodic constraint when reconnecting a sub-tree back to the parent tree. Fig. 11 shows an example of how the periodic constraint plays a role in the link cross deletion process. Specifically, in Fig. 11(a), we first delete link (B-C) that makes ⊥(B-S) isolated. To re-connect ⊥(B-S) back to the parent tree, we need to consider the periodic constraint. It is clear that the attempts in Figs. 11(b) and (c) are illegal to connect B as a child of S and G, respectively. The only legal graph is to connect B to F as in Fig. 11(d) since that does not violate the periodic constraint.

VI. SIMULATIONS AND RESULT ANALYSES

A. TEST CONDITIONS

We have considered 13 test networks of different sizes (from 14 to 200 nodes) for performance evaluation. These networks belong to real industrial projects of Huawei.[2] The numbers of network nodes and links are given in the first column (i.e., column $\boldsymbol{G} = (|\boldsymbol{N}|\,,|\boldsymbol{L}|)$) in Table I. According to the importance of each performance aspect, we set the weight factors for hop, link distance, long link, small angle, and link cross as 2.0, 5.0, 4.0, 2.0, and 2.0, respectively. Under this setting, we assume that minimizing the link distance (i.e., the aspects of *link distance* and *long link*) has a higher

[2] The test networks are real ones provided by Huawei. However, the detailed information of these networks is commercially privileged and the exact references for them cannot be provided.

TABLE 1. Performance comparison between ILP model and heuristic algorithm (single stage planning).

#	$G(\|N\|, \|L\|)$	Approach	|TH|	|LL|	|SA|	|LX|	|D|	|COST|	Gap
1	(14, 45)	ILP	21	4	4	1	42.19	278.95	2.77%
		Heu	21	4	2	3	43.734	286.67	
2	(15, 105)	ILP	23	2	1	1	7.971	97.855	0.89%
		Heu	23	2	1	1	8.146	98.73	
3	(16, 43)	ILP	44	8	1	0	124.38	743.9	1.95%
		Heu	42	8	2	0	127.68	758.4	
4	(16, 118)	ILP	25	1	0	0	6.37	85.85	0.00%
		Heu	25	1	0	0	6.37	85.85	
5	(16, 120)	ILP	25	3	0	0	9.91	111.55	3.23%
		Heu	25	2	0	2	10.63	115.15	
6	(24, 203)	ILP	62	3	0	0	97.71	624.55	0.64%
		Heu	61	3	0	0	98.91	628.55	
7	(27, 101)	ILP	101	9	1	0	185.06	1165.3	1.50%
		Heu	93	9	3	1	190.55	1182.75	
8	(49, 1176)	ILP	116	3	2	0	38.48	440.4	3.10%
		Heu	116	3	5	1	39.61	454.05	

priority because they are closely related to the whole system cost. However, it should be noted that our suggested approach would still be valid even if different combinations of weight factors are employed. The commercial software AMPL/Gurobi [23] was employed to solve the ILP model for small networks with MIPGAP = 0.00001. These were all solved to obtain optimal solutions within 10,000 seconds. The heuristic algorithms were implemented in Visual C++. In Step A of the heuristic algorithm, we considered 100 shuffled non-root node sequences for small networks ($|N| \leq 100$) and 50 shuffled non-root node sequences for large networks ($|N| > 100$) to choose a tree topology with the lowest ‘‘general cost’’ as the initial solution of Step A.

B. SINGLE STAGE PLANNING

Table I shows the results of the single stage case, in which in addition to the information of the test networks, columns |TH|, |LL|, |SA|, and |LX| correspond to the total numbers of traffic hops, long links, small angles, and links crosses, respectively. Column |D| shows the total distance of a planned tree topology in units of km and the column |COST| shows the ‘‘general cost’’ of each tree topology calculated based on (1). The last column ‘‘Gap’’ indicates the difference (in percentage) of ‘‘general cost’’ between the heuristic algorithm and the ILP model, which is calculated as $(Cost_{Heu} - Cost_{ILP})\,/Cost_{ILP}$. $Cost_{Heu}$ and $Cost_{ILP}$ are the ‘‘general costs’’ of the tree graphs planned by the heuristic algorithm and the ILP model, respectively.

As can be seen from the results of Table I, the overall performance of the proposed heuristic algorithm is very close to that obtained from the ILP model. In particular, for network 4, all of the performance aspects of the tree graphs planned by the heuristic algorithm and the ILP model are the same. For the remaining test networks, the performance differences for most aspects are small. This verifies the efficiency of our proposed heuristic algorithm.

As sample results, we also show the planned tree topologies obtained by the two approaches for networks 6 and 8 in Figs. 12 and 13, respectively, in which ‘‘ILP’’ corresponds to the topology obtained by the ILP model and ‘‘Heu’’ corresponds to the topology obtained by the heuristic algorithm. We can see that the topologies planned by the two approaches are close. This provides another perspective to verify the effectiveness of the heuristic algorithm.

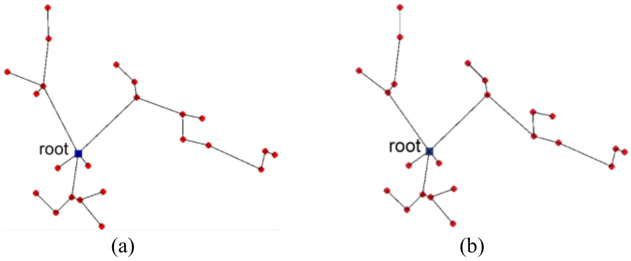


FIGURE 12. Tree graphs planned by the two approaches (network 6). (a) ILP. (b) Heu.

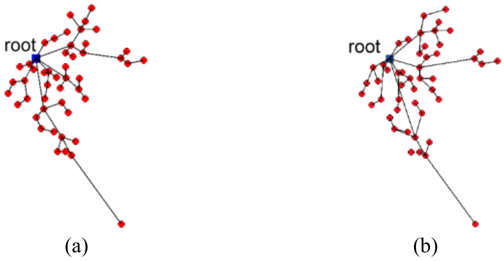


FIGURE 13. Tree graphs planned by the two approaches (network 8). (a) ILP. (b) Heu.

C. BENEFIT OF RE-OPTIMIZATION IN THE HEURISTIC ALGORITHM

The proposed heuristic algorithm consists of two steps, i.e., finding an initial solution (i.e., Step A) and re-optimizing the initial solution (i.e., Step B). In this section, we evaluate the benefit of the re-optimization step in the heuristic algorithm by comparing the performance of the heuristic algorithm that

TABLE 2. Result comparison between heuristic algorithms with and without re-optimization (single stage planning).

#	$G(\|N\|, \|L\|)$	Approach	|TH|	|LL|	|SA|	|LX|	|D|	|COST|	Gap
9	(56, 1053)	A	172	7	11	6	232.021	1566.105	26.25%
		A+B	210	3	7	2	141.012	1155.06	
10	(64, 1608)	A	164	7	12	17	681.692	3822.46	38.76%
		A+B	253	0	1	0	366.568	2340.84	
11	(133, 5560)	A	420	9	38	42	571.877	3895.385	32.36%
		A+B	481	3	16	12	320.978	2634.89	
12	(146, 7786)	A	499	17	39	68	1001.04	6285.2	43.30%
		A+B	552	3	7	8	483.574	3563.87	
13	(200, 17124)	A	656	12	51	81	1015.82	6703.1	39.50%
		A+B	753	5	25	21	486.425	4050.125	

TABLE 3. Result comparison between ILP model and heuristic algorithm (multi-stage planning).

#	$G(\|N\|, \|L\|)$	Approach	|TH|	|LL|	|SA|	|LX|	|D|	|COST|	Gap
1	(14, 45)	ILP	21	4	4	1	42.191	278.955	11.61%
		Heu.	21	5	2	3	47.87	311.35	
2	(15, 105)	ILP	23	2	1	1	7.971	97.855	0.89%
		Heu.	23	2	1	1	8.145	98.725	
3	(16, 43)	ILP	44	8	1	0	124.385	743.925	1.95%
		Heu.	42	8	2	0	127.684	758.42	
4	(16, 118)	ILP	25	1	0	0	6.368	85.84	0.00%
		Heu.	25	1	0	0	6.368	85.84	
5	(16, 120)	ILP	25	2	0	1	10.346	111.73	2.03%
		Heu.	25	2	1	1	10.4	114	
6	(24, 203)	ILP	62	3	0	0	97.706	624.53	0.17%
		Heu.	64	3	0	0	97.1161	625.5805	
7	(27, 101)	ILP	93	9	2	1	188.66	1171.3	0.00%
		Heu.	93	9	2	1	188.66	1171.3	

implements only Step A and the algorithm that implements both Steps A and B.

Table II shows the results of five large test networks with 56 to 200 nodes, in which various performance aspects are compared. It is evident from these results that the re-optimization step is necessary and efficient for the heuristic algorithm as it can significantly improve the performance of the planned tree topologies regarding "general cost." Moreover, as the network size increases, the performance improvement becomes even more significant. In the largest 200-node network, the re-optimization process can help reduce the total "general cost" by over 39.5% compared to the tree topology of the initial solution. Moreover, we also see that the re-optimization step can help improve each of the performance aspects. For example, for test networks 12 and 13, the re-optimization process can reduce the numbers of long links by 80% and 47%, respectively. Similarly, we can see that the numbers of small angles and link crosses are also significantly reduced after the re-optimization process.

D. IMPACT OF MULTI-STAGE CONSTRAINT

We also planned tree topologies for networks with the multi-stage constraint. Specifically, as sample studies, we divided all the non-root nodes into three stages based on their distances to the root node from the nearest to the farthest. Table III shows the results of the planned tree topologies.

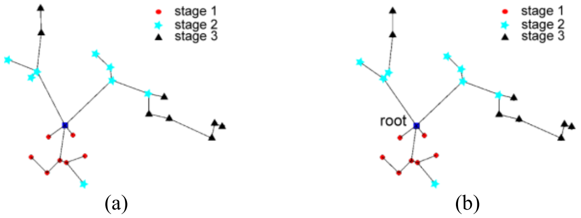


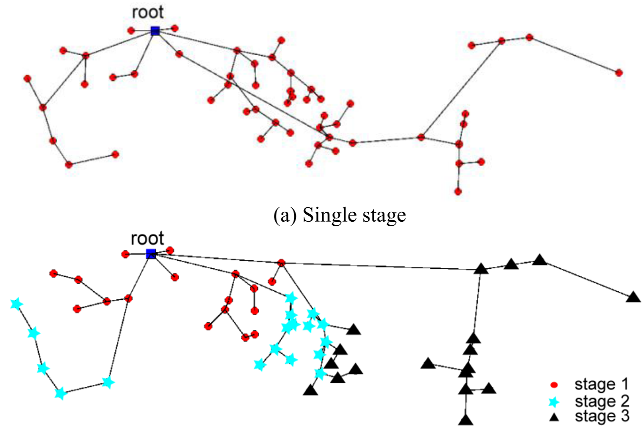
FIGURE 14. Tree topologies planned by the two approaches with the stage constraint (network 6). (a) ILP. (b) Heu.

We first compare the performance of the ILP model and the heuristic algorithm for the test networks where the number of nodes ranges from 14 to 27, i.e., from network 1 to network 7. As for the single stage case, we can see that the heuristic algorithm can give an overall performance which is close to that for the ILP model (the average performance gap is only 2.4%). In particular, for test networks 4 and 7, the heuristic algorithm achieves exactly the same performance as that of the ILP model. As a sample result, we also show the tree topologies planned by the ILP model and the heuristic algorithm for network 6 in Fig. 14. We can see that the two topologies are very similar.

Table IV shows the results of single stage (S-S) and multi-stage (M-S) planning for six larger test networks. Column "Gap" shows the cost difference between single stage and multi-stage planning, defined

TABLE 4. Result comparison between single stage and multi-stage planning.

#	$G(\|N\|, \|L\|)$	Approach	|TH|	|LL|	|SA|	|LX|	|D|	|COST|	Gap
8	(49, 1176)	S-S	116	3	5	1	39.61	454.05	2.87%
		M-S	121	3	5	1	40.2191	467.0955	
9	(56, 1053)	S-S	210	3	7	2	141.012	1155.06	-0.52%
		M-S	193	3	8	0	147.016	1149.08	
10	(64, 1608)	S-S	253	0	1	0	366.568	2340.84	0.03%
		M-S	256	0	1	0	365.511	2341.555	
11	(133, 5560)	S-S	481	3	16	12	320.978	2634.89	2.40%
		M-S	479	3	16	17	332.426	2698.13	
12	(146, 7786)	S-S	552	3	7	8	483.574	3563.87	1.13%
		M-S	549	2	8	12	491.661	3604.305	
13	(200, 17124)	S-S	753	5	25	21	486.425	4050.125	0.08%
		M-S	763	5	17	21	486.289	4053.445	



(a) Single stage

(b) Multi-stage

FIGURE 15. Comparison between single stage and multi-stage tree topology planning (network 9).

as $(Cost_{M-S} - Cost_{S-S})/Cost_{S-S}$, where $Cost_{S-S}$ and $Cost_{M-S}$ are the general costs of single stage and multi-stage planning, respectively. According to these results, we can see that though we now add the periodic constraint, the proposed heuristic algorithm can still plan a tree topology as efficiently as in the case without the constraint. Their average performance gap is no greater than 1.6%. As a sample result, Fig. 15 shows the planned tree topologies of the single stage and multi-stage scenarios for network 9. We can see that the two topologies are quite close. This therefore verifies the effectiveness of the proposed algorithm for a design scenario which is subject to the additional stage constraint.

VII. CONCLUSION

We focused on the microwave-based wireless backhaul network to plan its tree topology. Unlike the earlier studies that focused on just a single aspect or only a few optimization objectives, we jointly optimized multiple performance aspects subject to a variety of system constraints. To balance the different performance aspects of the optimization, we defined a "general cost" by assigning different weight factors to each of the performance aspects. An ILP model and an efficient heuristic algorithm were developed for the optimization problem. The planning schemes were also extended to support a practically realistic multi-stage scenario, which requires that an earlier-stage node should not be connected as a child to a later-stage node in a planned tree topology. Simulation results show that the heuristic algorithm is efficient and performs close to the ILP model for both the single stage and multi-stage design scenarios. Moreover, the proposed design approaches are effective in ensuring a multi-stage design which is close to its corresponding single stage design counterpart from the perspectives of the various design objectives considered.

ACKNOWLEDGMENT

Anliang Cai was with Soochow University, Suzhou, China. Parts of this paper were presented on Transparent Optical Networks 2014 [24] and Asia Communications and Photonics Conference 2014 [25].

REFERENCES

[1] A. Szlovencsak, I. Gódor, J. Harmatos, and T. Cinkler, "Planning reliable UMTS terrestrial access networks," *IEEE Commun. Mag.*, vol. 40, no. 1, pp. 66–72, Jan. 2002.

[2] M. Mahloo, P. Monti, J. Chen, and L. Wosinska, "Cost modeling of backhaul for mobile networks," in *Proc. ICC*, Jun. 2014, pp. 397–402.

[3] I. Gódor and G. Magyar, "Cost-optimal topology planning of hierarchical access networks," *Comput. Oper. Res.*, vol. 32, no. 1, pp. 59–86, Jan. 2005.

[4] J. Harmatos, "Planning of UMTS core networks," in *Proc. ISPIMRC*, Sep. 2002, pp. 740–744.

[5] J. Harmatos, A. Jüttner, and Á. Szentesi, "Cost–based UMTS transport network topology optimization," in *Proc. ICCC*, 1999, p. 1–9.

[6] A. Jüttner, A. Orbán, and Z. Fiala, "Two new algorithms for UMTS access network topology design," *Eur. J. Oper. Res.*, vol. 164, no. 2, pp. 456–474, Jul. 2005.

[7] Y. Wu and S. Pierre, "Optimization of 3G mobile network design using a hybrid search strategy," *J. Commun. Netw.*, vol. 7, no. 4, pp. 471–477, Dec. 2005.

[8] Y. Wu and S. Pierre, "Optimization of access network design in 3G networks," in *Proc. CCECE*, May 2003, pp. 781–784.

[9] U. Lauther, T. Winter, and M. Ziegelmann, "Proximity graph based clustering algorithms for optimized planning of UMTS access network topologies," in *Proc. ICT*, Feb./Mar. 2003, pp. 1329–1334.

[10] D. Tipper, T. Dahlberg, H. Shin, and C. Charnsripinyo, "Providing fault tolerance in wireless access networks," *IEEE Commun. Mag.*, vol. 40, no. 1, pp. 58–64, Jan. 2002.

[11] C. Charnsripinyo and D. Tipper, "Designing fault tolerant wireless access networks," in *Proc. MILCOM*, Oct. 2002, pp. 525–529.

[12] B. Dengiz, F. Altiparmak, and A. E. Smith, "Local search genetic algorithm for optimal design of reliable networks," *IEEE Trans. Evol. Comput.*, vol. 1, no. 3, pp. 179–188, Sep. 1997.

[13] J. Harmatos, A. Szentesi, and I. Gódor, "Planning of tree-topology UMTS terrestrial access networks," in *Proc. PIMRC*, 2000, pp. 353–357.

[14] J. Rong-Hong, "Design of reliable networks," *Comput. Oper. Res.*, vol. 20, no. 1, pp. 25–34, Jan. 1993.

[15] R. Nadiv and N. Tzvika, "Wireless backhaul topologies: Analyzing backhaul topology strategies," in *Proc. CWP*, Aug. 2010, pp. 1–15.

[16] F.-C. Kuo, F. A. Zdarsky, J. Lessmann, and S. Schmid, "Cost-efficient wireless mobile backhaul topologies: An analytical study," in *Proc. GLOBECOM*, Dec. 2010, pp. 1–5.

[17] M. St-Hilaire, "Topological planning and design of UMTS mobile networks: A survey," *Wireless Commun. Mobile Comput.*, vol. 9, no. 7, pp. 948–958, Jul. 2009.

[18] P. Monti, S. Tombaz, L. Wosinska, and J. Zander, "Mobile backhaul in heterogeneous network deployments: Technology options and power consumption," in *Proc. 14th ICTON*, Jul. 2012, pp. 1–7.

[19] A. Barradas, N. Correia, J. Coimbra, and G. Schütz, "Load adaptive and fault tolerant framework for energy saving in fiber-wireless access networks," *IEEE J. Opt. Commun. Netw.*, vol. 5, no. 9, pp. 957–967, Sep. 2013.

[20] V. Britvic, "Steps in UMTS network design," in *Proc. 12th MELECON*, May 2004, pp. 461–464.

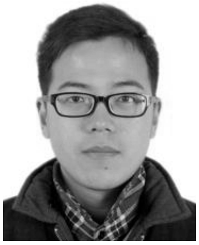
[21] T. Bu, M. C. Chan, and R. Ramjee, "Designing wireless radio access networks for third generation cellular networks," in *Proc. 24th INFOCOM*, Mar. 2005, pp. 68–78.

[22] Y. Wu and S. Pierre, "A new hybrid constraint-based approach for 3G network planning," *IEEE Commun. Lett.*, vol. 8, no. 5, pp. 277–279, May 2004.

[23] Gurobi. *Gurobi Optimizer Reference Manual*. [Online]. Available: http://www.gurobi.com

[24] Y. Li, G. Qiao, A. Cai, L. Shi, H. Zhao, and G. Shen, "Microwave backhaul topology planning for wireless access networks," in *Proc. 16th ICTON*, Jul. 2014, pp. 1–4.

[25] A. Cai, G. Qiao, Y. Li, L. Shi, and G. Shen, "Multi-period topology planning for microwave-based wireless backhaul networks," in *Proc. ACP*, 2014, paper AW3G.1.



YONGCHENG LI received the B.Sc. degree from Soochow University, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Soochow University. His research interests include optical network design, cloud computing, and optimization.

ANLIANG CAI received the B.Sc. degree in communication engineering from Hohai University, China, in 2011, and the M.Sc. degree in communication and information systems from Soochow University, China, in 2014. He is currently pursuing the Ph.D. degree with the City University of Hong Kong, Hong Kong. His research interests include optical networking, network design, and optimization.

GUANGYI QIAO received the master's degree in communication engineering from the University of Electronic Science and Technology of China. He was with the Advanced Research Department, Huawei in 2011. His research interests include network plan, evaluation, and optimization algorithm.

LEI SHI received the master's degree in computer science from Sichuan University. He was a Software Engineer with Huawei, where he was involved in OSN OTN PDU. In 2006, he became a member of the Advanced Research Department. His mainly interests include network plan, evaluation, and optimization algorithm.

SANJAY KUMAR BOSE (SM'91) received the B.Tech. degree from IIT Kanpur in 1976, and the master's and Ph.D. degrees from Stony Brook University, The State University of New York, USA, in 1977 and 1980, respectively. After working with the Corporate R&D Centre of the General Electric Company, Schenectady, NY, till 1982, he joined IIT Kanpur as an Assistant Professor, where he became a Professor in 1991. He left IIT Kanpur in 2003 to join as a Faculty Member with the School of EEE, NTU, Singapore. In 2008, he left NTU to join IIT Guwahati, where he is currently a Professor with the Department of EEE, and the Dean of Alumni Affairs and External Relations. He has been working in various areas in the field of computer networks and queuing systems, and has published extensively in the area of optical networks and network routing. He is a fellow of IETE (India), and a member of Sigma Xi and Eta Kappa Nu.

GANGXIANG SHEN (S'98–M'99–SM'12) received the B.Eng. degree from Zhejiang University, China, the M.Sc. degree from Nanyang Technological University, Singapore, and the Ph.D. degree from the University of Alberta, Canada, in 2006. He is a Distinguished Professor with the School of Electronic and Information Engineering, Soochow University, China. Before he joined Soochow University, he was a Lead Engineer with Ciena, Linthicum, MD. He was also an Australian Post-Doctoral Fellow with the University of Melbourne. His research interests include integrated optical and wireless networks, spectrum efficient optical networks, and green optical networks. He has authored or co-authored more than 55 peer-reviewed technical papers. He is a Lead Guest Editor of the *IEEE Journal on Selected Areas in Communications* Special Issue entitled Next-Generation Spectrum Efficient and Elastic Optical Transport Networks. He is an Editorial Board Member of *Optical Switching and Networking*. He received the Young Researcher New Star Scientist Award in the 2010 Scopus Young Researcher Award Scheme in China. He was a recipient of an Izaak Walton Killam Memorial Award from the University of Alberta and a Canadian NSERC Industrial R&D Fellowship.