Computers and Operations Research 81 (2017) 102–118

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Exact and heuristic approaches based on noninterfering transmissions for joint gateway selection, time slot allocation, routing and power control for wireless mesh networks[☆]

Kagan Gokbayrak[a,*], E. Alper Yıldırım[b,1]

[a] Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey
[b] Department of Industrial Engineering, Koç University, Rumelifeneri Yolu, 34450 Sarıyer, İstanbul, Turkey

ARTICLE INFO

Article history:
Received 27 November 2014
Revised 28 September 2016
Accepted 28 September 2016
Available online 3 October 2016

Keywords:
Mixed integer linear programming
Heuristic method
k-opt hill climbing
Wireless mesh network

ABSTRACT

Wireless mesh networks (WMNs) provide cost-effective alternatives for extending wireless communication over larger geographical areas. In this paper, given a WMN with its nodes and possible wireless links, we consider the problem of gateway node selection for connecting the network to the Internet along with operational problems such as routing, wireless transmission capacity allocation, and transmission power control for efficient use of wired and wireless resources. Under the assumption that each node of the WMN has a fixed traffic rate, our goal is to allocate capacities to the nodes in proportion to their traffic rates so as to maximize the minimum capacity-to-demand ratio, referred to as the service level. We adopt a time division multiple access (TDMA) scheme, in which a time frame on the same frequency channel is divided into several time slots and each node can transmit in one or more time slots. We propose two mixed integer linear programming formulations. The first formulation, which is based on individual transmissions in each time slot, is a straightforward extension of a previous formulation developed by the authors for a related problem under a different set of assumptions. The alternative formulation, on the other hand, is based on sets of noninterfering wireless transmissions. In contrast with the first formulation, the size of the alternative formulation is independent of the number of time slots in a frame. We identify simple necessary and sufficient conditions for simultaneous transmissions on different links of the network in the same time slot without any significant interference. Our characterization, as a byproduct, prescribes a power level for each of the transmitting nodes. Motivated by this characterization, we propose a simple scheme to enumerate all sets of noninterfering transmissions, which is used as an input for the alternative formulation. We also introduce a set of valid inequalities for both formulations. For large instances, we propose a three-stage heuristic approach. In the first stage, we solve a partial relaxation of our alternative optimization model and determine the gateway locations. This stage also provides an upper bound on the optimal service level. In the second stage, a routing tree is constructed for each gateway node computed in the first stage. Finally, in the third stage, the alternative optimization model is solved by fixing the resulting gateway locations and the routing trees from the previous two stages. For even larger networks, we propose a heuristic approach for solving the partial relaxation in the first stage using a neighborhood search on gateway locations. Our computational results demonstrate the promising performance of our exact and heuristic approaches and the valid inequalities

1. Introduction

A wireless mesh network (WMN) comprises a finite number of radio nodes that are capable of communicating with one another and with the nearby clients in a wireless fashion. A small portion of these radio nodes are designated to be gateways with wired connections to the Internet to enable the flow of traffic into and out of the WMN. The rest of the nodes forward traffic toward and from gateways in a multi-hop fashion through other radio nodes. By eliminating the need to install a wired connection to each node,

☆ This work was supported in part by TÜBITAK (Turkish Scientific and Technological Research Council) grant 110M312.

* Corresponding author.
E-mail addresses: kgokbayr@bilkent.edu.tr (K. Gokbayrak), alperyildirim@ku.edu.tr (E.A. Yıldırım).

[1] This author was supported in part by TÜBITAK (Turkish Scientific and Technological Research Council) grant 112M870 and by TÜBA-GEBİP (Turkish Academy of Sciences Young Scientists Award Program).

http://dx.doi.org/10.1016/j.cor.2016.09.021
0305-0548/

WMNs provide a cost-effective alternative to extend the coverage of communication networks to larger geographic regions.

Since the transmission medium is shared by all nodes, interference among simultaneous transmissions is a major concern in wireless communications. The interference effect on the unintended receiver is highly dependent on how far the interferer is and on the strength of its transmitted signal. In this paper, we adopt the physical interference model in [13] based on the signal-to-interference-plus-noise ratio (SINR). In this model, transmission from node i to node j is deemed to be successful (in the sense that node j can correctly decode the signal from node i) if the ratio of the strength of the signal from node i received at node j to the total strength of all signals received from all other transmitting nodes at node j plus the ambient noise is above a certain threshold value. In contrast to simpler interference models based on identification of interfering node pairs, the SINR scheme is more realistic since the cumulative effect of all simultaneous transmissions is taken into account in this model.

Unlike ad hoc networks, where nodes may be battery operated, the nodes of WMNs have constant power supplies and hence do not have energy conservation concerns. However, there is still a need for power control for interference prevention purposes. The strength of the signal received at a destination node can be increased by increasing the power level of the transmitting node. However, such an increase in the power level also increases the signal strength received at unintended nodes, causing additional interference if they are destination nodes for other transmissions. Therefore, in the SINR model, power levels of transmitting nodes should be determined judiciously in order to have successful simultaneous transmissions on several links, referred to as spatial reuse.

The level of spatial reuse, i.e., the maximum number of simultaneous transmissions, is generally not adequate to form a connected network. As a remedy, nodes can share the wireless capacity by either transmitting on different frequency channels or taking turns transmitting on the same frequency channel. In the former approach, a node can be equipped with multiple radios so that it can simultaneously send or receive signals on different channels without any interference. In this paper, however, we adopt the latter approach, namely, the time division multiple access (TDMA) scheme. In this scheme, a time frame is divided equally into T time slots. Nodes continuously store incoming traffic and forward it over their wireless links during the allocated time slots. Note that the capacity of a wireless link is directly proportional to the number of time slots during which it is activated. The frame structure is then repeated in a periodic manner.

As we are not addressing the coverage problem for wireless access networks (e.g., as in [2]) in which nodes are installed at candidate positions to cover clients, we assume that the network topology is given. Since the nodes aggregate traffic flows for a large number of clients, we assume that each node in the WMN has a fixed traffic rate that should be forwarded to the Internet. Under a limited budget for a given number of gateways, our objective is to maximize the service level, which we define to be the smallest ratio of the allocated capacity to the demand of each node. A service level value larger than one implies that the resulting WMN can continue to satisfy the traffic demand of each node at least for a while under the assumption that traffic rate of each node grows proportionally over time. In that sense, we search for the network design that will satisfy the demand for the longest period.

To eliminate the need for reordering of packets at the destination, we assume that the traffic of each node is carried to a gateway on a single path as in [22]. Even though multi-path routing has the benefit of load balancing, the additional protocol overhead can be significant. Specifically, we adopt the destination based routing scheme, i.e., one routing tree is constructed for each gateway node. The TDMA time slots are to be allocated to the wireless links on the routing trees so that they have enough capacity to accommodate the traffic flow.

In this paper, our main objective is to design and operate a WMN so as to maximize the service level. We select the gateway locations, form a routing tree for each one of these gateways, and determine the number of time slots that should be allocated to each noninterfering transmission set on these routing trees. We also determine the power level of each transmitting node to enable spatial reuse of the time slots.

First, we propose a mixed integer linear programming formulation, which is a straightforward extension of a previous formulation developed by the authors for a related problem under a different set of assumptions [12]. In this formulation, in order to model power control and interference, we need decision variables and constraints defined for each time slot. As the number of time slots in a frame is generally large, these models can get too large to be solved to optimality. Therefore, in an attempt to develop an alternative formulation, we identify simple necessary and sufficient conditions to have multiple links of the WMN activated in the same time slot. These conditions can easily be applied as a preprocessing step to enumerate all possible noninterfering sets of transmissions in a given WMN. Furthermore, our characterization yields a power level for each of the transmitting nodes as a byproduct. Motivated by this observation, we develop an alternative mixed integer linear programming (MILP) formulation based on sets of noninterfering transmissions. This formulation allows us to completely eliminate power control and interference issues from consideration. We also redefine decision variables and modify constraints so that the problem size is independent of the number of time slots in a frame.

We perform computational experiments on WMNs with different characteristics. Our computational results reveal that our alternative formulation based on noninterfering link sets significantly outperforms the first formulation in terms of the solution quality and running time. For smaller and simpler networks, our alternative model can usually compute an exact solution in a short amount of time. For larger and more complicated networks, solving even the alternative model to optimality becomes computationally challenging due to the increase in the numbers of nodes, links, and noninterfering transmission sets. We therefore propose a three-stage heuristic approach for such instances. In the first stage, we determine the gateway locations by solving a partial relaxation of our alternative optimization model. The second stage consists of constructing a routing tree by fixing the resulting gateway locations computed in the first stage. Finally, in the third stage, we solve the alternative optimization problem by fixing the gateway locations and the routing trees computed in the previous two stages. Note that, in this final stage, we eliminate a considerable number of noninterfering link sets, which allows us to quickly compute a good feasible solution.

The objective value of the partially relaxed problem in the first stage yields an upper bound that we utilize in evaluating the performance of our three-stage heuristic method. For even larger networks in which even the partial relaxation in the first stage can be difficult to solve, we propose a neighborhood search scheme on gateway locations for computing a local optimal solution. Our computational results demonstrate the effectiveness of our heuristic approaches.

This paper is organized as follows: We discuss related literature in the next section. In Section 3, we present our first mixed integer linear programming formulation. Then, we present simple necessary and sufficient conditions in order to have successful simultaneous transmissions on any given subset of links. Our characterization also yields an appropriate power level for each of the transmitting nodes as a byproduct. Then, we propose a simple procedure to enumerate all sets of noninterfering transmissions and develop an alternative optimization model based on such sets. A set of valid inequalities is also introduced in this section. In Section 4, we present heuristic solution approaches for large networks. Section 5 is devoted to the presentation of our computational experiments on four sample networks. In this section, we evaluate the performances of our optimization models and the heuristic approaches. We also compare the proposed heuristic with the heuristic from [12] in this section. We conclude the paper with some remarks in Section 6.

2. Related work

After the turn of the century, there has been a tremendous research effort on WMNs. We discuss below only a few papers related to our problem. For other references, we refer the reader to survey papers, e.g., [1], [20], and [6].

There are quite a few papers on the gateway selection problem. A common objective is to minimize the number of gateways under quality-of-service requirements on delay and bandwidth. Chandra et al. [8] showed that it is NP-hard to determine the smallest number of gateways to satisfy node demands. Bejerano [5] and Aoun et al. [3] presented the gateway selection problem as variants of the capacitated facility location problem with additional constraints for routing and link capacities. They both presented polynomial-time heuristic methods that yield near-optimal results.

Some of the more recent studies employed the physical interference model as we do in this paper. Papadaki and Friderikos [17] considered the link scheduling problem and developed a mixed integer linear programming (MILP) model to maximize the number of transmissions within a number of time slots while guaranteeing that each link transmitted at least once. They proposed an approximate dynamic programming methodology for this NP-hard problem. Quintas and Friderikos [21] considered the minimum power scheduling problem and formulated an MILP model to minimize the total transmission power within a number of time slots while guaranteeing that all transmission requests were satisfied. Assuming a fixed power level at all nodes and with predetermined gateways, Badia et al. [4] formulated an MILP model for the joint routing and link scheduling problem and proposed a genetic algorithm to solve this NP-hard problem.

The concept of noninterfering transmissions can be found in some recent papers. Karnik et al. [14] defined noninterfering transmissions, referred to as independent set of links, for a given power vector and determined a loose upper bound for the maximum size of independent sets to limit the number of checks for complete enumeration performed on the set of all subsets of the links. Luo et al. [16], on the other hand, developed a complete enumeration algorithm based on the proposition that any subset of an independent set is also independent and determined independent sets of increasing sizes until the largest sets were obtained. Note that these two papers do not adjust transmission powers to generate noninterfering links, they merely present the complete list for a given power setting. Luo et al. [16] considered the objective of maximizing the minimum throughput for a given set of gateways. They determined the percentage of the time an independent set should be active. For large networks, they proposed a column generation approach to limit the number of independent sets in the formulation by employing exact or greedy pricing. This paper also provided some engineering insights, illustrating, for instance, that multipath routing does not produce a significant improvement in performance over single path routing and that there is a diminishing gain of spatial reuse. The latter argument was supported with the following observation: For a large number of randomly generated WMNs where more than six simultaneous noninterfering transmissions were possible, employing only noninterfering transmissions of sizes less than or equal to three yielded an almost optimal throughput. Capone et al. [7] considered the joint problem of scheduling, routing, power control, and rate adaptation for WMNs so as to minimize the number of time slots needed to deliver the traffic between pairs of nodes. Similar to our problem formulation, they associated integer decision variables to noninterfering transmissions, referred to as configurations in [7]. Since the number of configurations increased exponentially with the network size, they applied a column generation method to solve a continuous relaxation of the problem to obtain a lower bound. For an upper bound, they solved the original problem using only the configurations with positive values in an optimal solution of the relaxed problem.

The following two papers are most closely related to the work presented in this paper. Targon et al. [22] and the authors of this paper [12] presented MILP formulations for the joint optimization of gateway placement, routing, and link scheduling to support a given set of node demands. The objective in [22] was to minimize total gateway costs and they employed source-based single-path routing. On the other hand, the service level was maximized under tree-based routing in [12].

In this paper, we extend our work in [12] as follows: First, in our earlier paper, a single transmission power level for each node was to be determined, i.e., whenever node i transmitted in any time slot, it was only allowed to transmit at its designated power level $\pi_i \in [0, 1]$, which was a decision variable that represents the ratio of the power level to the maximum power level. In this paper, however, we assume that nodes can transmit at different power levels in different time slots. Employing this new setting, we can derive rules and power levels to form all possible noninterfering transmission sets. Second, while our first formulation in this study is a straightforward extension of the formulation in [12] to the current setting, our alternative MILP formulation is based on noninterfering transmission sets, rather than on individual transmissions. Moreover, since it is the number of allocated time slots that determines the capacity on a wireless link, the alternative formulation employs integer decision variables for the total number of allocated slots to each noninterfering transmission set in a frame, rather than binary assignment variables for each individual transmission in each time slot. Consequently, the size of the alternative formulation is independent of the number of time slots in a frame. Finally, based on this alternative formulation, we propose new heuristic methods for gateway selection and routing.

3. Problem definition and formulations

We consider a wireless mesh network (WMN) composed of N nodes. We are interested in forwarding traffic from the nodes of the WMN to the Internet through gateway nodes. We assume that node i has an uplink traffic rate of d_i, $i = 1, \ldots, N$. Any node of the WMN can be designated as a gateway node by equipping it with a wired link of data rate of a, connecting the WMN to the Internet. We assume that we have a budget for selecting G gateway nodes from among N nodes of the WMN, where $G < N$. The remaining nodes of the WMN can forward traffic toward the gateway nodes in a wireless multi-hop fashion through other nodes.

We assume that each node has an isotropic antenna that distributes power equally to a spherical region. Hence, the received power is inversely proportional to the square of the distance between the transmitter and the receiver nodes, if they are both located in free space. If nodes are located on irregular terrain, however, due to reflection, refraction, diffraction and absorption, the path loss exponent typically ranges between 3 and 3.4. The study [10] presents a model for path loss exponent, which is dependent on the height of the base station antenna. In our examples, we assume a path loss exponent value of 3, i.e., the signal transmitted by node i is received at node j at a power level of $l_{ij} = Kr_{ij}^{-3}$ times the transmitted signal power, where l_{ij} and r_{ij} are the path loss value and the distance, respectively, between nodes i and j. In free space, the multiplier K would be the ratio of the receiving surface area times the antenna gain to 4π. On irregular terrain, though, one needs to solve a parabolic equation that includes earth's curvature, refraction index of air, etc. (see [23]) to calculate path loss. For simplicity, we assume $K = 1$ in our examples and emphasize the path loss due to distance only.

In order to model wireless communication from node i to node j, we use the signal-to-interference-plus-noise ratio (SINR) scheme (see, e.g., [13]). In this model, node j can successfully decode a signal from node i if, at node j, the ratio of the power of the signal received from node i to the power of the signals from all other transmitting nodes plus the ambient noise level is above a certain threshold value. More specifically, if each transmitting node n transmits at a power level of P_n, then the signal from node i can be correctly decoded by node j if and only if

$$\frac{P_i \, l_{ij}}{\eta_j + \sum_{\substack{n=1 \\ n \notin \{i,j\}}}^{N} P_n \, l_{nj}} \geq \gamma_c, \tag{1}$$

where η_j denotes the ambient noise level at node j and γ_c denotes the SINR threshold value for a transmission rate of c. Note that higher transmission rates require higher threshold values, but since we assume a single rate of c, we can drop the subscript in our notation and employ γ instead.

We assume that node i can transmit at a maximum power of $P_i^{\max}$, $i = 1, \ldots, N$. Henceforth, we will normalize the power levels by defining

$$\rho_i = \frac{P_i}{P_i^{\max}} \in [0, 1], \quad i = 1, \ldots, N. \tag{2}$$

We also denote the maximum power that can be received at node j for signals transmitted by node i as

$$g_{ij} = P_i^{\max} l_{ij}, \quad 1 \leq i \leq N; \ 1 \leq j \leq N; \ i \neq j. \tag{3}$$

Then, it follows that the physical interference model in (1) can equivalently be expressed as

$$\frac{\rho_i \, g_{ij}}{\eta_j + \sum_{\substack{n=1 \\ n \notin \{i,j\}}}^{N} \rho_n \, g_{nj}} \geq \gamma. \tag{4}$$

A node pair (i, j) is defined as a wireless link if the ratio of the power of the signal received from node i, which is transmitting at maximum power level, to the ambient noise level η_j at node j exceeds the signal-to-interference-plus-noise ratio (SINR) threshold γ. The set of directed wireless links of this network is therefore given by

$$E = \{(i, j) | P_i^{\max} \, l_{ij} \geq \gamma \, \eta_j\} = \{(i, j) | g_{ij} \geq \gamma \, \eta_j\}. \tag{5}$$

Adopting the time division multiple access (TDMA) scheme, we divide the wireless link capacity c into T equal-data-rate parts, assuming that there are T time slots in a frame. Spatial reuse is possible, i.e., multiple transmissions can be enabled in any time slot as long as these transmissions do not interfere with each other. We also assume the half-duplex operation of nodes, i.e., in the same time slot, a node can either transmit to at most one other node or receive from at most one other node.

We assume that a destination-based routing scheme is employed, i.e., for each gateway node, a routing tree will be constructed in such a way that each node of the WMN belongs to exactly one routing tree. It follows that the traffic from each non-gateway node will be forwarded to a gateway node on a single path of wireless links in the WMN.

In this paper, we are interested in designing and operating a WMN in which we select the gateway nodes, construct routing trees for each gateway node, determine the number of time slots in which each wireless link will be activated to form these routing trees along with the power level of each transmitting node so as to maximize the service level, which is defined as the smallest ratio of the allocated capacity to the demand of each node. We represent the above problem with two different mixed integer linear programming (MILP) models. We list all the parameters of the problem in Table 1.

Table 1
Problem parameters.

N	Number of nodes
E	Set of wireless links
d_i	Uplink traffic rate of node i, $i = 1, \ldots, N$
G	Number of gateway nodes
l_{ij}	Path loss between nodes i and j, $i = 1, \ldots, N;\ \ j = 1, \ldots, N; i \neq j$
γ	SINR threshold
η_j	Ambient noise level at node j, $j = 1, \ldots, N$
$P_i^{\max}$	Maximum power level of node i, $i = 1, \ldots, N$
g_{ij}	$P_i^{\max} l_{ij}$, $1 \leq i \leq N;\ 1 \leq j \leq N;\ i \neq j$
c	Data rate of a wireless link
a	Data rate of a wired link at a gateway node
T	Number of time slots in a frame

In the next subsection, we present an optimization model, adapted from the one presented in [12] to the current problem under consideration. In Section 3.2, we present simple necessary and sufficient conditions in order to have noninterfering transmissions on any given subset of wireless links in the same time slot. In Section 3.3, we use this characterization to describe an effective procedure for enumerating the set of all noninterfering transmissions. Finally, we present an alternative optimization formulation based on the set of noninterfering transmissions in Section 3.4.

3.1. First optimization model

In this section, we present our first optimization model, denoted by (WMN0), which is a straightforward extension of the optimization model proposed in [12]. First, we present our decision variables in Table 2.

Note that the decision variable w represents the service level. The variables f_{ij} and ϕ_i are the flow variables. The time slot assignment variables are represented by v_{ij}^t and u_{ij}. The variables z_{ij} are related to routing. Finally, y_i and ρ_i^t denote the gateway selection and the power control variables, respectively.

Our first MILP model (WMN0) is presented below:

$$\max w \tag{6}$$

subject to

$$w d_i + \sum_{h:(h,i) \in E} f_{hi} = \phi_i + \sum_{j:(i,j) \in E} f_{ij}, \quad i \in \{1, \ldots, N\}, \tag{7}$$

$$\phi_i \leq a y_i, \quad i \in \{1, \ldots, N\}, \tag{8}$$

$$f_{ij} \leq \frac{c}{T} u_{ij}, \quad (i, j) \in E, \tag{9}$$

$$\sum_{j:(i,j) \in E} z_{ij} = 1 - y_i, \quad i \in \{1, \ldots, N\}, \tag{10}$$

$$\sum_{i=1}^{N} y_i = G, \tag{11}$$

$$u_{ij} = \sum_{t=1}^{T} v_{ij}^t, \quad (i, j) \in E, \tag{12}$$

$$u_{ij} \leq T z_{ij}, \quad (i, j) \in E, \tag{13}$$

$$\sum_{k:(k,i) \in E} v_{ki}^t + \sum_{j:(i,j) \in E} v_{ij}^t \leq 1, \quad i \in \{1, \ldots, N\}, t \in \{1, \ldots, T\}, \tag{14}$$

Table 2
Decision variables of (WMN0).

w	:	the minimum ratio of the allocated capacity to the demand at each node, i.e., the service level
f_{ij}	:	the traffic flow on link $(i, j) \in E$
ϕ_i	:	the traffic flow exiting the WMN at node $i \in \{1, 2,..., N\}$
v_{ij}^t	=	$\begin{cases} 1, & \text{if wireless link } (i,j) \text{ is active at time slot } t, \\ 0, & \text{otherwise,} \end{cases} \quad (i,j) \in E, t \in \{1,\dots,T\}$
u_{ij}	:	the number of time slots in which the link $(i, j) \in E$ is active
z_{ij}	=	$\begin{cases} 1, & \text{if wireless link } (i,j) \text{ belongs to a routing tree,} \\ 0, & \text{otherwise,} \end{cases} \quad (i,j) \in E$
y_i	=	$\begin{cases} 1, & \text{if node } i \text{ is a gateway,} \\ 0, & \text{otherwise,} \end{cases} \quad i \in \{1,\dots,N\}$
ρ_i^t	:	the ratio of the power transmitted by node i at time slot t to the maximum power $P_i^{\max}$, $i \in \{1,\dots,N\}, t \in \{1,\dots,T\}$

$$\rho_i^t \leq \sum_{j:(i,j)\in E} v_{ij}^t, \quad i \in \{1,\dots,N\}, t \in \{1,\dots,T\}, \tag{15}$$

$$\rho_i^t g_{ij} + M_{ij}(1 - v_{ij}^t) \geq \gamma\eta_j + \gamma \sum_{\substack{n=1 \\ n\neq i,j}}^{N} \rho_n^t g_{nj},$$

$$(i,j) \in E, t \in \{1,\dots,T\}, \tag{16}$$

where

$$M_{ij} = \gamma\left(\eta_j + \sum_{\substack{n=1 \\ n\neq i,j}}^{N} g_{nj}\right), \quad (i,j) \in E,$$

$$v_{ij}^t \in \{0,1\}, \quad (i,j) \in E, t \in \{1,\dots,T\}, \tag{17}$$

$$y_i \in \{0,1\}, \quad \in \{1,\dots,N\}, \tag{18}$$

$$z_{ij} \in \{0,1\}, \quad (i,j) \in E, \tag{19}$$

$$\rho_i^t \geq 0, \quad \in \{1,\dots,N\}, t \in \{1,\dots,T\}, \tag{20}$$

$$\phi_i \geq 0, \quad i \in \{1,\dots,N\}, \tag{21}$$

$$f_{ij} \geq 0, \quad (i,j) \in E, \tag{22}$$

$$u_{ij} \geq 0, \quad (i,j) \in E, \tag{23}$$

$$w \geq 0. \tag{24}$$

Our objective is to maximize the service level w. Under the assumption that the traffic demand of each node will increase proportionally over time, this objective serves to design the network that can last the longest without any upgrades. Note that the current demand of each node will be satisfied if and only if the service level is at least one. The flow balance at each node is captured by the constraints (7). The constraints (8) represent the gateway capacity constraints at each gateway node. Note that the data rate of a wireless link is given by c. Under the TDMA scheme, a time frame is divided into T time slots and we assume that each time slot has an equal data rate given by c/T. Therefore, the constraints (9) imply that the total flow on link (i, j) cannot exceed the allocated wireless capacity. Each gateway node routes the incoming traffic to the Internet using a wired connection. For each node that is not a gateway, the constraints (10) ensure that each such node is allowed to route its traffic to exactly one other node in a wireless fashion. The number of gateways is specified by (11) and the number of time slots assigned to each link is given by the constraints (12). Any wireless link that is not on any route cannot be activated in any time slot by (13). Due to half-duplex operation, a node cannot transmit and receive in the same time slot by the constraints (14). By (15), the power level at each node is zero when there is no transmission on any of the wireless links incident at that node. Note that, due to (14), the right-hand side of (15) is at most one. The signal-to-interference-plus-noise ratio (SINR) scheme given by (1) is formulated in (16) in the form of a big-M constraint, and a possible value that the parameter M can take is given. The domains of all decision variables are specified in (17)–(24).

An optimal solution of the optimization model (WMN0) yields the gateway locations, routing, assignments of time slots to subsets of wireless links, and the power schedules so as to maximize the service level. Note that (WMN0) consists of $2NT + |E|T + 3N + 3|E| + 1$ constraints, $T|E| + |E| + N$ binary and $NT + N + 2|E| + 1$ nonnegative continuous decision variables. As we have multiplicative terms with T, for realistic network sizes, the size of (WMN0) becomes very large. Therefore, we exploit the structure of the physical interference constraint (1) to obtain an alternative model for the same problem.

3.2. Noninterfering transmission sets

Note that, due to half-duplex operation, it is not possible to have simultaneous transmissions on any two wireless links that share a common node in the same time slot. Therefore, noninterfering transmission sets can only consist of subsets of node-disjoint links.

It follows from the physical interference model in (1) and the definition of E in (5) that each wireless link $(i, j) \in E$ by itself can be activated in a time slot by simply setting $\rho_i = \gamma\eta_j/g_{ij}$ and $\rho_k = 0$ for each $k = 1,\dots,N,\ k \neq i$.

Our next result gives a simple and complete characterization in order for a set of at least two node-disjoint links to have noninterfering transmissions in the same time slot.

Proposition 1. *Let $L = \{(k_1, l_1), (k_2, l_2), \dots, (k_m, l_m)\}$ be a set of node-disjoint links in a wireless mesh network, where $m \geq 2$. Then, all of the links in L can be activated in the same time slot if and only if the following system has a feasible solution:*

$$A\rho = b, \quad \rho > 0, \quad \rho \leq e, \tag{25}$$

where $A \in \mathbb{R}^{m\times m}$ is given by

$$A_{ij} = \begin{cases} g_{k_i,l_i}, & if \quad i = j, \\ -\gamma g_{k_j,l_i}, & otherwise, \end{cases} \quad i = 1,\dots,m; \quad j = 1,\dots,m,$$

$b \in \mathbb{R}^m$ is given by $b_i = \gamma\,\eta_{l_i},\ i = 1,\dots,m,\ \rho \in \mathbb{R}^m$ denotes the vector of decision variables $\rho_i,\ i = 1,\dots,m$ defined as in (2), g_{ij} is defined as in (3), and $e \in \mathbb{R}^m$ denotes the vector of all ones.

Proof. Let $\hat{\rho} \in \mathbb{R}^m$ be a feasible solution of (25). If we set $\rho_{k_j} = \hat{\rho}_j,\ j = 1,\dots,m$, and $\rho_k = 0$ for each $k \in \{1,\dots,N\}\backslash\{k_1,\dots,k_m\}$, it is easy to verify that each of the inequalities (4) corresponding to each of the links in L will be satisfied. Therefore, all of the links in L can be activated in the same time slot.

Conversely, suppose that all of the links in L can be activated in the same time slot. It follows from (4), the definitions of A and b, and $\rho_k \in [0, 1]$ for each $k = 1, \ldots, N$ that the following system has a feasible solution:

$$A\rho \geq b, \quad \rho > 0, \quad \rho \leq e.$$

To show the existence of a feasible solution that satisfies $A\rho = b$, let us consider the following linear programming problem:

$$\min\{e^T\rho : A\rho \geq b, \quad \rho \geq 0, \quad \rho \leq e\}.$$

Note that this problem has a nonempty feasible region by our assumption. Furthermore, any feasible solution ρ should satisfy

$$g_{k_i,l_i}\rho_i - \gamma \sum_{j=1,\ j\neq i}^{m} g_{k_j,l_i}\rho_j \geq \gamma\,\eta_{l_i}, \quad i = 1, \ldots, m,$$

which, combined with the feasibility requirement that $\rho > 0$, implies that

$$\rho_i > \frac{\gamma\,\eta_{l_i}}{g_{k_i,l_i}}, \quad i = 1, \ldots, m. \tag{26}$$

Since the feasible region of the given linear programming problem is bounded, there exists at least one optimal solution $\rho^* \in \mathbb{R}^m$. Note that $\rho^* > 0$ by (26). We claim that $A\rho^* = b$. Suppose, for a contradiction, that there exists $r \in \{1, \ldots, m\}$ such that $(A\rho^*)_r > b_r$, i.e.,

$$g_{k_r,l_r}\rho_r^* - \gamma \sum_{j=1, j\neq r}^{m} g_{k_j,l_r}\rho_j^* > \gamma\,\eta_{l_r}.$$

In this case, if we decrease ρ_r^* until the strict inequality above is satisfied with equality while fixing all other components of ρ^*, it is easy to verify that the resulting solution still satisfies the other constraints of $A\rho \geq b$ since the coefficient of ρ_r is negative in each of the remaining rows of A. It follows that the new solution has a strictly smaller objective function value than that of ρ^*, which contradicts the optimality of ρ^*. Therefore, ρ^* is a solution of the system (25). □

The next result establishes a useful property of the matrix A defined in Proposition 1.

Lemma 2. *Let $L = \{(k_1, l_1), (k_2, l_2), \ldots, (k_m, l_m)\}$ be a set of node-disjoint links in a wireless mesh network, where $m \geq 2$. Suppose that the following system has a feasible solution:*

$$A\rho = b, \quad \rho > 0, \tag{27}$$

where A and b are defined as in Proposition 1. Then, A is nonsingular and the unique solution is given by $\rho = A^{-1}b$.

Proof. Suppose that the system (27) has a feasible solution. By a similar argument as in the proof of Proposition 1, any feasible solution should satisfy (26), i.e.,

$$\rho_i > \frac{\gamma\,\eta_{l_i}}{g_{k_i,l_i}} = \ell_i, \quad i = 1, \ldots, m.$$

Let $\hat{\rho} > 0$ be such an arbitrary feasible solution of the system (27). Suppose, for a contradiction, that A is singular, i.e., there exists a nonzero $d \in \mathbb{R}^m$ such that $Ad = 0$. Without loss of generality, we can assume d has at least one positive component. Note that if all components of d are nonpositive, we can replace d with $-d$. Let $\mu > 0$ be defined as $\mu = \min_{j=1,\ldots,n: d_j>0} \frac{\hat{\rho}_j - \ell_j}{d_j}$. Then, $\hat{\rho} - \mu d > 0$ and $A(\hat{\rho} - \mu d) = b$, which implies that $\hat{\rho} - \mu d$ is a feasible solution of (27). Note that this solution has at least one component that does not satisfy the inequality in (26). This is a contradiction. □

Combining Proposition 1 and Lemma 2, we obtain the following simple and useful characterization.

Corollary 3. *Let $L = \{(k_1, l_1), (k_2, l_2), \ldots, (k_m, l_m)\}$ be a set of node-disjoint links in a wireless mesh network, where $m \geq 2$. Then, all of the links in L can be activated in the same time slot if and only if A is nonsingular and $0 \leq A^{-1}b \leq e$, where A and b are defined as in Proposition 1.*

Proof. By Proposition 1, if all of the links in L can be activated in the same time slot, then (25) has a feasible solution. Note that the feasibility of (25) implies the feasibility of (27). It follows from Lemma 2 that A is nonsingular and the unique solution satisfies $0 < A^{-1}b \leq e$.

Conversely, if A is nonsingular and $0 \leq A^{-1}b \leq e$, then $A^{-1}b$ is a feasible solution of (25). Then, all of the links in L can be activated in the same time slot by Proposition 1. □

This corollary presents necessary and sufficient conditions in order for a set of wireless links to be activated simultaneously. We can also employ this corollary to determine a set of power settings for the source node of each of these wireless links. If A is a nonsingular matrix, then the unique solution of the system $A\rho = b$ is given by

$$\rho_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, \ldots, m, \tag{28}$$

where $A_i \in \mathbb{R}^{m\times m}$ is the matrix obtained from A by replacing the ith column of A with the right-hand side vector b. If these links are noninterfering, each ρ_i should be positive and less than or equal to one. Using this observation, the next two corollaries establish necessary and sufficient conditions for noninterfering wireless link sets of size two and size three, respectively.

Corollary 4. *Given a WMN, two node-disjoint links (i, j) and (s, d) can be activated together in the same time slot if and only if*

$$g_{ij}g_{sd} - \gamma^2 g_{id}g_{sj} \geq \gamma \max\left\{g_{sd}\eta_j + \gamma g_{sj}\eta_d,\ g_{ij}\eta_d + \gamma g_{id}\eta_j\right\}. \tag{29}$$

Furthermore, in this case, the power settings for the source nodes are given by

$$\rho_i = \frac{\gamma g_{sd}\eta_j + \gamma^2 g_{sj}\eta_d}{g_{ij}g_{sd} - \gamma^2 g_{id}g_{sj}},$$

$$\rho_s = \frac{\gamma g_{ij}\eta_d + \gamma^2 g_{id}\eta_j}{g_{ij}g_{sd} - \gamma^2 g_{id}g_{sj}}.$$

Following a similar discussion, in [12], we introduced and demonstrated the benefits of the following set of valid inequalities that prevent an interfering pair of links from being activated in the same time slot

$$v_{ij}^t + v_{sd}^t \leq 1, \quad t \in \{1, \ldots, T\}, \tag{30}$$

whenever the edges (i, j) and (s, d) do not satisfy the condition (29). In this paper, we utilize this condition in Corollary 4 in a different way: We use it to determine pairs of links that can be activated in the same time slot. As the prescribed power setting for a source node depends on the other link in the noninterfering pair, we remark that a result similar to Corollary 4 cannot be established under the assumptions of [12].

Similar conditions and power settings for subsets with three or more links can be derived using Corollary 3 and the relation (28). However, these characterizations get too complicated to be included in the paper due to space considerations. Rather, Corollary 3 gives us a simple and efficient computational recipe to check if any given set of node-disjoint links can be activated in the same time slot. The power settings for the source nodes of these links can be derived using the relation (28).

3.3. Enumeration of noninterfering transmission sets

In this section, given a WMN, we develop a simple computational method for enumerating all noninterfering transmission sets. Our method consists of two stages. In the first stage, we compute the largest number of links that can be activated in the same time slot. We utilize a simple optimization model for this task. The resulting size serves as a terminating condition for the second stage. The second stage consists of the repeated application of Corollary 3 on carefully selected subsets of node-disjoint links of size less than or equal to the threshold value computed in the first stage. Similar to [16], we exploit the proposition that any proper subset of a noninterfering link set is also noninterfering. Note that the terminating condition prevents unnecessary checks for larger sets of links. In addition, we do not need to prespecify power settings. Rather, our approach yields an appropriate power level for each transmitting node as a byproduct.

For the first stage, we propose a mixed integer linear programming (MILP) problem, denoted by (AUX), for computing the largest number of links that can be activated in the same time slot. The parameters of this problem are already defined in Table 1, and the decision variables are presented in Table 3.

We next present our MILP model (AUX) that computes the largest number of links that can be activated in the same time slot.

$$\max \sum_{(i,j)\in E} s_{ij} \tag{31}$$

subject to

$$\sum_{k:(k,i)\in E} s_{ki} + \sum_{j:(i,j)\in E} s_{ij} \le 1, \qquad i = 1, \dots, N, \tag{32}$$

$$\rho_i \le \sum_{j:(i,j)\in E} s_{ij}, \qquad i = 1, \dots, N, \tag{33}$$

$$\rho_i g_{ij} + M_{ij}(1 - s_{ij}) \ge \gamma \eta_j + \gamma \sum_{n=1 n\notin\{i,j\}}^{N} \rho_n g_{nj}, \quad (i,j) \in E, \tag{34}$$

where

$$M_{ij} = \gamma \left(\eta_j + \sum_{\substack{n=1 \\ n\notin\{i,j\}}}^{N} g_{nj} \right), \quad (i,j) \in E,$$

$$s_{ij} \in \{0, 1\}, \quad (i,j) \in E, \tag{35}$$

$$\rho_i \ge 0, \quad = 1, \dots, N. \tag{36}$$

The constraint set (32) ensures that a node can either transmit to at most one other node or receive from at most one other node in the same time slot. If a node does not transmit a signal in a time slot, then the constraint set (33) sets its power level to zero. Finally, the constraint set (34) is used to formulate the SINR interference model.

The MILP model (AUX) has a total of $2N + |E|$ constraints, $|E|$ binary and N nonnegative continuous decision variables. Our computational experiments reveal that (AUX) is fairly easy to solve for networks of reasonable sizes. Note that any optimal solution of (AUX) yields a set of noninterfering transmissions with the largest cardinality.

In the second stage, we start by populating the set $\mathcal{S}$ of noninterfering link sets with the set $\{(i, j)\}$ for each link (i, j) in the set E. These sets constitute the noninterfering sets of cardinality one, which we denote by $\mathcal{S}_1$. Then, we determine the noninterfering sets of larger sizes $\mathcal{S}_m$, where $m \ge 2$, as follows: In order to determine if a node-disjoint link set of cardinality m is noninterfering, we first check if all of its subsets of cardinality $m-1$ are noninterfering. If they are, then we apply Corollary 3 to see if the set itself is noninterfering and determine the appropriate power values. We continue in this fashion until we determine all noninterfering sets of node-disjoint links of size less than or equal to the optimal value of (AUX). We therefore obtain a complete list of all sets of noninterfering transmissions together with the corresponding power settings that enable such transmissions. A pseudocode of the algorithm is given in Algorithm 1.

Algorithm 1: Enumeration of all noninterfering transmissions.

```
Data: Problem parameters in Table 1
Result: Set S of all noninterfering transmissions
Solve the optimization problem (AUX) and let κ denote its
optimal value;
S ← ∅;
for each (i, j) ∈ E do
 | S ← S ∪ {(i, j)};
end
m ← 2;
while m ≤ κ do
    for each node-disjoint subset F ⊆ E of size m do
        if each subset F′ ⊆ F of size m − 1 belongs to S then
            Apply Corollary 3 to the set of links in F;
            if conditions of Corollary 3 are satisfied then
             | S ← S ∪ F;
            end
        end
    end
    m ← m + 1;
end
```

In Algorithm 1, the most computationally intensive operation is the verification of the conditions of Corollary 3, which has a worst-case complexity of $O(m^3)$ for each subset $F \subseteq E$ of size m because a system of equations involving an $m \times m$ matrix needs to be solved. Since there are at most $\binom{|E|}{m}$ node-disjoint subsets $F \subseteq E$ of size m and since $m \le \kappa$, the number of such verifications is bounded above by $\sum_{m=2}^{\kappa} \binom{|E|}{m} \le \sum_{m=2}^{\kappa} |E|^m = O(|E|^{\kappa+1})$. It follows that the worst-case complexity of Algorithm 1 is $O(|E|^{\kappa+1}\kappa^3)$. Our computational results reveal that the running time of Algorithm 1 is fairly negligible even for considerably large instances.

3.4. Alternative optimization model

In this section, we develop an alternative MILP model for the problem of maximizing the service level of a given WMN. As a first step, in contrast to defining time slot assignment variables over the set of edges E as in (WMN0), we define them over the set of all noninterfering transmission sets $\mathcal{S}$, which can be completely enumerated using Algorithm 1 in Section 3.3. In other words, we

Table 3
Decision variables of (AUX).

s_{ij}	=	$\begin{cases} 1, & \text{if link } (i,j) \text{ is included in the set of noninterfering links,} \\ 0, & \text{otherwise,} \end{cases} \quad (i,j) \in E$
$\rho_i \in [0, 1]$	:	the ratio of transmission power of node i to the maximum power level denoted by $P_i^{\max}$, $i = 1, \dots, N$

replace decision variables v with $\bar{x}$ defined as follows:

$$\bar{x}_\sigma^t = \begin{cases} 1, & \text{if set } \sigma \text{ is active at time slot } t, \\ 0, & \text{otherwise,} \end{cases} \quad \sigma \in \mathcal{S}, t \in \{1, \dots, T\}.$$

Consequently, since each of these sets in $\mathcal{S}$ satisfies the constraints (14)–(16), we can replace those constraints with the following constraint:

$$\sum_{\sigma \in \mathcal{S}} \bar{x}_\sigma^t = 1, \quad t \in \{1, \dots, T\}. \tag{37}$$

This constraint suggests that we can activate only one noninterfering set at each time slot t. We can also remove the power control decision variables ρ as we have already obtained them for each element of $\mathcal{S}$.

We also need to update the constraint (12) according to the new variable definition. Utilizing the subset of $\mathcal{S}$ defined as $\mathcal{S}_{ij} = \{\sigma \in \mathcal{S} | (i, j) \in \sigma\}$, which is the set of noninterfering sets that include the link (i, j), we write

$$u_{ij} = \sum_{\sigma \in \mathcal{S}_{ij}} \sum_{t=1}^{T} \bar{x}_\sigma^t, \quad (i, j) \in E, \tag{38}$$

Note that there are $|\mathcal{S}|T$ binary decision variables $\bar{x}_\sigma^t$ and there are T equality constraints in (37). In order to eliminate the number of time slots T in the number of constraints and decision variables, as a second step, we replace binary decision variables $\bar{x}_\sigma^t$ with integer decision variables x_σ defined as

$$x_\sigma = \sum_{t=1}^{T} \bar{x}_\sigma^t, \quad \sigma \in \mathcal{S}. \tag{39}$$

Note that x_σ represents the number of time slots in which the noninterfering transmission set σ is active in a TDMA frame. Then, employing the definition of x_σ in (37) and (38), the alternative MILP model, denoted by (WMN1), can be formulated as follows:

$$\max w \tag{40}$$

subject to (7)–(11), (13), (18), (19), (21)–(24) and

$$\sum_{\sigma \in \mathcal{S}} x_\sigma = T, \tag{41}$$

$$u_{ij} = \sum_{\sigma \in \mathcal{S}_{ij}} x_\sigma, \quad (i, j) \in E, \tag{42}$$

$$x_\sigma \geq 0, \quad \text{integer}, \quad \sigma \in \mathcal{S}. \tag{43}$$

The constraint (41), which is simply obtained by summing (37) over all time slots t, states that the total number of time slots allocated to noninterfering transmission sets should be equal to the number of available time slots in a TDMA frame. Note that it is possible for a noninterfering transmission set to be activated in several time slots. The number of time slots in which each link is active is given by (42), which replaces the constraints (12) in (WMN0). The integer decision variables of (WMN1) are defined in (43).

An optimal solution of (WMN1) yields the gateway locations, the routing trees, and the number of time slots allocated to each noninterfering transmission set in a TDMA frame so as to maximize the service level. Note again that the power level of each transmitting node in any noninterfering transmission set is already determined using Corollary 3 and the relation (28). Therefore, the power management issue is completely eliminated in this alternative formulation.

The original model (WMN0) and the alternative model (WMN1) are compared in terms of the number of constraints and the number of variables in Table 4.

Table 4
Sizes of two models (WMN0) and (WMN1).

Model	(WMN0)	(WMN1)
# of constraints	$2NT + \lvert E\rvert T + 3N + 3\lvert E\rvert + 1$	$3N + 3\lvert E\rvert + 2$
# of nonnegative continuous variables	$NT + N + 2\lvert E\rvert + 1$	$N + 2\lvert E\rvert + 1$
# of binary variables	$\lvert E\rvert T + N + \lvert E\rvert$	$N + \lvert E\rvert$
# of nonnegative integer variables	0	$\lvert \mathcal{S}\rvert$

Note that (WMN1) has a smaller size than that of (WMN0), except for the number of integer variables that can grow exponentially as a function of the number of edges $|E|$. The number of constraints and the number of binary and continuous variables in (WMN1) are both linear in the number of nodes N and the number of edges $|E|$. In contrast with (WMN0), we remark that the size of (WMN1) is independent of the number of time slots T. Therefore, it is the preferred model when T is large. In Section 4, we exploit this feature of (WMN1) to develop a three-stage heuristic method, which is based on initially solving the problem under the assumption that T goes to infinity so as to determine decent gateway locations.

3.5. A set of valid inequalities

As mentioned above, for a given number of gateways, the demand of each node in a WMN can be satisfied if and only if the service level w is at least one. Under this assumption, we propose a set of valid inequalities for the optimization models in this section.

In both models, the inequality set (13) ensures that links that are not on any routing tree are not activated. On the other hand, if a link (i, j) is on a routing tree, then this link has to carry at least the traffic d_i of node i toward a gateway. Since the capacity of the link (i, j) in a time slot is given by c/T, it follows that the link (i, j) has to be allocated at least $\lceil \frac{d_i T}{c} \rceil$ time slots to carry node i's own traffic toward a gateway. Hence, assuming that the service level w is at least one, we propose the following set of valid inequalities:

$$u_{ij} \geq \left\lceil \frac{d_i T}{c} \right\rceil z_{ij}, \quad (i, j) \in E, \tag{44}$$

We discuss the effect of these valid inequalities in Section 5.

4. A three-stage heuristic method

Note that the number of nonnegative integer variables of (WMN1) may grow exponentially as a function of the number of edges $|E|$. Therefore, on large networks, (WMN1) can be a very challenging model for MILP solvers. In this section, we propose a three-stage heuristic approach for computing an approximate solution of (WMN1) on larger networks.

In the first stage, we determine a set of gateways and an upper bound for the optimal service level. In the second stage, we determine routing trees rooted at those gateways. Finally, in the third stage, we fix the gateway locations and the corresponding routing trees in the original optimization model (WMN1) so as to allocate time slots to noninterfering transmission sets.

4.1. Stage one: upper bound computation and gateway selection

In the first stage, our goals are to determine the gateway locations and to obtain a good upper bound on the optimal service level. In order to achieve both of these objectives, we seek a partial relaxation of the original optimization model (WMN1). The optimal value of this partially relaxed model will serve as an upper bound for assessing the quality of our heuristic solutions. Our main goal is to strike a balance between the computational cost of solving the resulting relaxation and the quality of the upper bound given by the optimal value of this relaxation.

We first consider the extreme case, i.e., the linear programming (LP) relaxation. Under the assumption that the wired link rate a is at least as large as the wireless link rate c, in any optimal solution of the LP relaxation, each node acts as a fractional gateway and transfers its traffic directly to the Internet through wired links. Missing the wireless side of the problem, the LP relaxation does not provide a good upper bound on the optimal service level. In an attempt to obtain a better bound by precluding fractional gateways, we formulate below a partial relaxation of (WMN1), where we keep the gateway selection variables $\{y_i\}_{i=1}^N$ binary so that each of the non-gateway nodes must communicate through wireless links to forward traffic to a gateway node.

We remove the constraint sets (10) and (13), and the routing variables z_{ij} in (19) from (WMN1). In other words, we do not enforce single path routing. Moreover, we relax the integrality condition on the variables x_σ in (WMN1). Then, we can further perform a change of variable by defining $\hat{x}_\sigma = x_\sigma / T$ for each $\sigma \in \mathcal{S}$. In a similar fashion, the decision variables u_{ij} can be replaced by $\hat{u}_{ij} = u_{ij}/T$ for each $(i, j) \in E$. The new variables $\hat{x}_\sigma$ and $\hat{u}_{ij}$ can be interpreted as the fraction of a TDMA frame in which the noninterfering set σ and the wireless link (i, j) are active, respectively. The modified optimization model, denoted by (WMN-S1), determines the best set of gateways under multipath routing and under the assumption that any arbitrary fraction of a TDMA frame can be allocated to any noninterfering transmission set. Therefore, the service level obtained from (WMN-S1) is an upper bound on the optimal service level obtained from (WMN1).

We next present our partial relaxation, denoted by (WMN-S1):

$$\max w \tag{45}$$

subject to (7), (8), (11), (18), (21), (22), (24), and

$$f_{ij} \le c\,\hat{u}_{ij}, \quad (i, j) \in E, \tag{46}$$

$$\sum_{\sigma \in \mathcal{S}} \hat{x}_\sigma = 1, \tag{47}$$

$$\hat{u}_{ij} = \sum_{\sigma \in \mathcal{S}_{ij}} \hat{x}_\sigma, \quad (i, j) \in E, \tag{48}$$

$$\hat{u}_{ij} \ge 0, \quad (i, j) \in E, \tag{49}$$

$$\hat{x}_\sigma \ge 0, \quad \sigma \in \mathcal{S}. \tag{50}$$

Constraints (9), (41), and (42) are modified using the new variable definitions as (46), (47), and (48), respectively. Similarly, variable definitions (49) and (50) replace earlier definitions (23) and (43), respectively. Note that the model (WMN-S1) does not depend on the number of time slots T. Therefore, the gateway set and the upper bound obtained from this partial relaxation are independent of the number of time slots in a TDMA frame.

Unfortunately, for large networks, it may still take a long time to solve (WMN-S1). To address this issue, we propose a neighborhood search method that can be used to find an approximate solution.

4.1.1. *k-opt hill climbing for gateway selection*

In [11], Fischetti and Lodi introduce the notion of "local branching." In this scheme, given a binary vector $\bar{y} \in \mathbb{R}^N$ and a positive integer $k \le N$, the authors define a k-opt neighborhood as the set of all binary vectors which differ from $\bar{y}$ in at most k components. Local branching is proposed as a branching criterion, in which, given an incumbent solution $\bar{y}$, the solution space is partitioned into two sets with respect to solutions that are in the k-opt neighborhood of $\bar{y}$ and the remaining solutions.

We adopt the same k-opt neighborhood definition for the binary vectors $y \in \mathbb{R}^N$ in (WMN-S1) corresponding to gateway locations. By (11), exactly G of the N components of $y \in \mathbb{R}^N$ should be equal to one in any feasible solution. Let us denote the set of all such binary vectors by Γ. Let us also define $A(\bar{y}) = \{i \in \{1, \ldots, N\} : \bar{y}_i = 1\}$, i.e., the set $A(\bar{y})$ consists of the indices of the gateway nodes represented by the solution $\bar{y}$. The corresponding k-opt neighborhood of $\bar{y}$ is then defined by

$$\mathcal{N}_k(\bar{y}) := \left\{ y \in \Gamma : \sum_{i \in A(\bar{y})} y_i \ge G - k \right\}.$$

Note that $\hat{y} \in \mathcal{N}_k(\bar{y})$ if and only if the corresponding gateway sets $A(\hat{y})$ and $A(\bar{y})$ have at least $G - k$ common elements, where $k \in \{1, \ldots, G\}$. As the parameter k increases, additional binary vectors from Γ are added to the set of neighbors, i.e., $\mathcal{N}_k(\bar{y}) \subset \mathcal{N}_{k+1}(\bar{y})$ for all $k \in \{1, \ldots, G-1\}$, and $\mathcal{N}_G(\bar{y}) = \Gamma$.

We propose the following local-branching-based approach for computing a heuristic solution of the first stage model (WMN-S1). Starting at an initial binary vector $\bar{y} \in \Gamma$ with the corresponding gateway set $A(\bar{y})$, we seek the best solution of (WMN-S1) only among the solutions in the k-opt neighborhood of $\bar{y}$, i.e., we solve (WMN-S1) with the following additional constraint:

$$\sum_{i \in A(\bar{y})} y_i \ge G - k. \tag{51}$$

If the gateway set obtained in the solution of (WMN-S1) with the additional constraint (51) is different from the starting gateway set $A(\bar{y})$, we update the incumbent binary vector and the corresponding gateway set and repeat this procedure until the set of gateway nodes does not change. Note that the optimal value obtained in each iteration is not lower than the optimal value at the previous iteration and the set of gateways is always updated to another gateway set corresponding to one of the k-opt neighbors of the previous incumbent binary vector. We therefore refer to this local search as the k-opt hill climbing method.[2]

We remark that, for larger instances, the k-opt hill climbing method can be employed as an alternative to directly solving the first stage optimization model (WMN-S1). The k-opt hill climbing can be started from a randomly selected initial set of gateways. Once we obtain the final set of gateways, which clearly depends on the initial set, we can proceed with the next two steps of our three-stage heuristic approach. Since the final solution of the three-stage heuristic will depend on the initial gateway selection, we can repeat this approach for several randomly selected gateway sets and report the best solution. We discuss the quality of the resulting solutions from (WMN-S1) and the k-opt hill climbing method in Section 5.

For small values of k, our computational experiments in Section 5 indicate that the optimization model (WMN-S1) with the additional constraint (51) can usually be solved fairly quickly. However, since the number of neighbors would also be small, the algorithm may get stuck at a local optimal solution of (WMN-S1). A larger value of k would bring additional neighbors into consideration at the expense of longer computation times. Numerical examples in Section 5 demonstrate this trade-off.

Note that, unless we choose $k = G$, the k-opt hill climbing method does not guarantee global optimality. Therefore, in contrast with directly solving (WMN-S1), we cannot obtain an upper bound for the service level using this approach.

[2] Note that a similar k-opt (also called k-change in [18]) local search method was proposed in [9] and [15] as a heuristic solution method for the Traveling Salesman Problem (TSP), where a local search was performed over the set of all tours that were different in at most k edges from the current tour.

4.2. Stage two: tree-based routing

We modify (WMN-S1) by adding the constraints (10) and (13), and the routing variables z_{ij} in (19). We also fix the gateway locations computed in the first stage, i.e., we set $y_i = \bar{y}_i, i = 1, \ldots, N$, where $\bar{y}$ denotes the optimal (or near-optimal) values of y in (WMN-S1). We therefore obtain the second optimization model, denoted by (WMN-S2), presented below:

$$\max w$$

subject to (7), (19), (21), (22), (24), (46)–(50), and

$$\phi_i \le a\bar{y}_i, \quad i = 1, \ldots, N, \tag{52}$$

$$\sum_{j:(i,j)\in E} z_{ij} = 1 - \bar{y}_i, \quad i = 1, \ldots, N, \tag{53}$$

$$\hat{u}_{ij} \le z_{ij}, \quad (i,j) \in E. \tag{54}$$

The constraints (52) and (53) are the gateway capacity (8) and the single path (10) constraints, respectively, that are reformulated for the gateway locations given by $\bar{y}$. The constraint set (54), on the other hand, is the constraint set (13) expressed in terms of the variables $\hat{u}$. Note that we no longer need the gateway budget constraints (11), as the gateway locations are already given. An optimal solution of (WMN-S2) yields routing trees rooted at the gateway nodes obtained from the first stage.

4.3. Stage three: time slot allocation

We impose additional constraints in the original model (WMN1) by setting the values of the variables y and z to their respective optimal (or near-optimal) values, denoted by $\bar{y}$ and $\bar{z}$, from the first two stages. In other words, we fix them in the original optimization model (WMN1) in order to compute the allocation of time slots to noninterfering transmission sets. Fixing the routing tree in (WMN1) eliminates a rather large number of noninterfering transmission sets in $\mathcal{S}$ from consideration. Therefore, the resulting MILP problem can usually be solved very quickly, even for large networks.

We present below the MILP formulation, denoted by (WMN-S3), that performs the time slot allocation:

$$\max w$$

subject to (7), (9), (21)–(24), (41)–(43), (52), and

$$u_{ij} \le T\bar{z}_{ij}, \quad (i,j) \in E. \tag{55}$$

Note that the constraint set (55) is the constraint set (13) reformulated for the routing trees given by $\bar{z}$. In this model, we do not need single path constraints (10), gateway budget constraints (11), and the variable definitions in (18) and (19) as the gateway locations and the routing trees are already determined.

In the next section, we present our computational results on networks of different sizes and characteristics in order to demonstrate the effectiveness of our heuristic approaches.

5. Computational results

In this section, we consider four networks denoted by A, B, C, and D. Networks A and C are of grid topology consisting of node configurations of 5 × 6 and 7 × 7, respectively, and the vertical and horizontal distances between any two neighboring nodes are assumed to be 1 kilometer (km). Networks B and D, on the other hand, consist of 27 nodes and 37 nodes, respectively, that are randomly generated from a uniform distribution over a 4 km-by-4 km square.

In each of these networks, each gateway connection is assumed to have a capacity of $a = 45$ Mbps as of a T3 cable connection and each link has a capacity of $c = 24$ Mbps for the SINR threshold value of $\gamma = 50$. Note that this value of γ guarantees a 100% frame reception ratio under static path loss (see [19]).

Table 5
Sizes of the four sample networks.

Label	N	$\lvert E\rvert$
A	30	98
B	27	162
C	49	168
D	37	254

Table 6
Some parameter values.

SINR threshold	$\gamma = 50$
Ambient noise level at node j, $j = 1, \ldots, N$	$\eta_j = 1.5 \times 10^{-10}$ mW
Maximum power level of node i, $i = 1, \ldots, N$	$P_i^{\max} = 15$ mW
Data rate of a wireless link	$c = 24$ Mbps
Data rate of a wired link at a gateway node	$a = 45$ Mbps

The ambient noise η is assumed to be 1.5×10^{-10} mW at all nodes. In each of the four networks, each node is assumed to have a maximum transmission power of 15 mW. The path loss l_{ij} is assumed to be $l_{ij} = r_{ij}^{-3}$, where r_{ij} denotes the distance between node i and node j. Note that under these parameter settings, a node with a maximum transmission power of 15 mW can form wireless links toward nodes that are at most 1.26 km away. The resulting wireless links for Networks A, B, C, and D are presented in Fig. 1. We remark that each wireless link actually represents two directed links in reverse directions due to the identical power settings for the nodes. Table 5 summarizes the sizes of each of the four networks, where N denotes the number of nodes and E denotes the set of directed wireless edges.

We summarize the values of our parameters in Table 6.

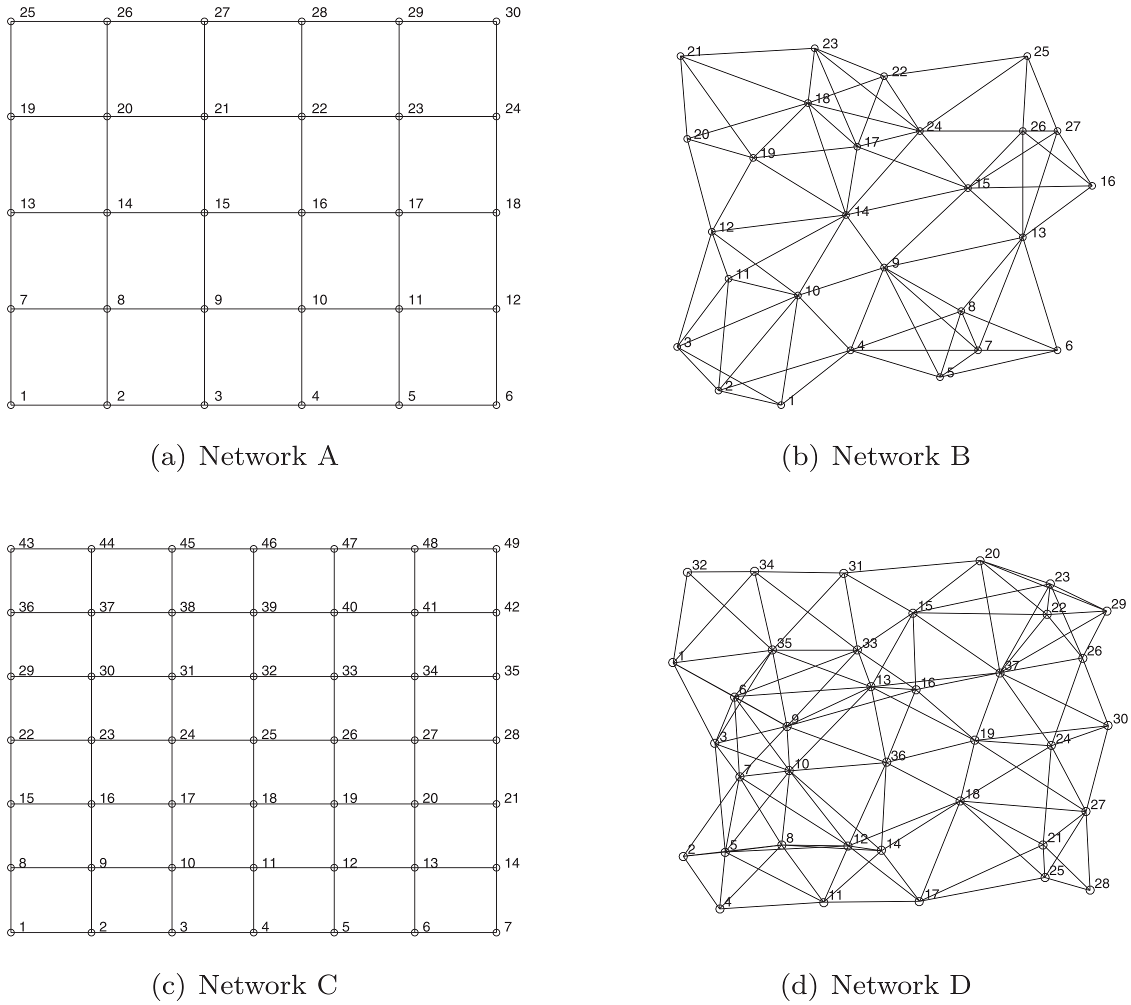
In each network, the demand of each node (in Mbps) was randomly chosen uniformly from the set $\{0.1, 0.2, \ldots, 1.0\}$. Table 7 presents the demand of each node in each network.

Our computational experiments were performed on a dual 2.4 GHz Intel Xeon E5620 CPU PC with 16 GB RAM. The optimization problems were solved using CPLEX 12.6 in parallel mode using up to 16 threads. We set a CPU time limit of 20 h (72,000 s) for each optimization problem unless stated otherwise. Algorithm 1 was implemented in Matlab R2015b, whereas the three-stage heuristic employing k-opt in the first stage was implemented in Java.

In our experiments, in order to obtain service levels exceeding one, the number of gateways G takes values from the sets {3, 4, 5, 6}, {2, 3}, {4, 5, 6, 7}, and {3, 4, 5} for Networks A, B, C, and D, respectively. For each of the four sample networks, we consider the instances where the number of time slots T in a frame takes values from the set {64, 128}.

For each of the four networks, we first solved the optimization model (AUX) in order to compute the maximum number of noninterfering wireless links. Next, we used Algorithm 1 to enumerate all noninterfering transmission sets in each of the four networks. For each network, the maximum number of noninterfering wireless links and the total number of noninterfering transmission sets are presented in Table 8 along with the number of noninterfering transmission sets of sizes one through five. In Table 8, $\mathcal{S}_m$ denotes the set of noninterfering transmission sets of size m, $m = 1, 2, \ldots, 5$. The CPU times (in seconds) for solving (AUX) and implementing Algorithm 1 are also presented.

Table 8 indicates that the CPU time required for solving the optimization model (AUX) and implementing Algorithm 1 is less than half a minute on each of the four networks. Therefore, the maximum number of noninterfering wireless links and the set of all noninterfering transmission sets can be computed fairly effectively even for considerably large instances.



(a) Network A (b) Network B

(c) Network C (d) Network D

Fig. 1. Example network topologies.

5.1. Model comparisons

We start by comparing the proposed model (WMN1) against the extension of the previous model (WMN0) supplemented with the valid inequalities (30) proposed in [12]. The comparisons are performed on all four networks.

We compare the solutions obtained within 72,000 s of CPU time in Table 9, which is organized as follows: The first set of columns presents the details of each instance. Note that the number of gateways is denoted by G and the number of time slots in a TDMA frame is denoted by T. The second set and the third set of columns present the objective function value of the best feasible solution and the best upper bound computed within the time limit, the corresponding optimality gap, and the CPU time using (WMN0) with the set of valid inequalities (30) and (WMN1), respectively. In two instances for which CPLEX ran out of memory (indicated by an asterisk), we report the CPU time when the process was killed. Note that all CPU times in this section are reported in seconds and the optimality gap is defined as the difference between the best upper bound and the objective function value of the best feasible solution divided by the latter. The boldface entries in each row indicate an improvement over the other model.

For the simplest network, Network A, (WMN1) could solve seven instances out of eight to optimality within the time limit. The best solutions, best upper bounds, and optimality gaps obtained by (WMN1) were better than those obtained from (WMN0) with (30) in all eight instances.

For Network B, (WMN1) could compute an optimal solution in two out of four instances. The best solutions of (WMN0) with (30) in all instances were inferior to the best solutions obtained from (WMN1). Moreover, the optimality gaps obtained from the former were larger in all instances.

In all instances of Network C, (WMN0) with (30) returned optimality gaps that were over 350%. Even though (WMN1) computed better solutions in all these instances, it could not close the optimality gap in any of them. However, the optimality gaps from (WMN1) were considerably lower than those from (WMN0) with (30). We remark that, in two instances, (WMN1) ran out of memory.

One of the instances of Network D was solved to optimality by (WMN1). In the other five instances, the optimality gaps obtained from (WMN1) were always lower than 8%. (WMN0) with (30), on the other hand, returned inferior results with optimality gaps over 250% in all instances.

Overall, our results on four networks reveal that (WMN1) outperformed (WMN0) supplemented with (30) in all of the instances. Now that we have shown the benefit of modeling with noninterfering sets over the whole time frame as opposed to individual links in individual time slots, we proceed with our computational results using only the model (WMN1).

5.2. The effect of the valid inequality (44)

To demonstrate the effects of the set of valid inequalities in (44) on (WMN1), we solve (WMN1) with and without them on

Table 7
Node demands in Mbps.

Node	Network A	Network B	Network C	Network D
1	0.9	0.1	0.6	0.5
2	1.0	0.3	0.6	0.4
3	0.2	0.1	0.1	0.6
4	1.0	0.1	0.4	0.6
5	0.7	0.9	0.7	0.9
6	0.1	0.7	0.4	0.8
7	0.3	0.4	0.3	0.7
8	0.6	1.0	0.3	0.4
9	1.0	0.1	0.3	0.9
10	1.0	0.5	0.9	0.6
11	0.2	0.4	0.8	0.4
12	1.0	0.8	0.2	1.0
13	1.0	0.8	0.3	0.9
14	0.5	0.2	0.4	0.6
15	0.9	0.5	0.3	0.7
16	0.2	0.5	0.2	0.6
17	0.5	0.7	0.6	0.8
18	1.0	0.8	0.2	0.4
19	0.8	0.8	0.2	1.0
20	1.0	0.3	0.3	0.5
21	0.7	0.7	0.3	0.3
22	0.1	0.7	0.9	0.9
23	0.9	0.2	0.6	0.9
24	1.0	0.2	0.6	0.8
25	0.7	0.5	0.9	0.9
26	0.8	1.0	0.4	0.8
27	0.8	0.4	0.7	0.5
28	0.4		0.4	0.4
29	0.7		0.3	1.0
30	0.2		0.3	0.5
31			0.3	1.0
32			0.1	0.2
33			0.8	1.0
34			0.2	1.0
35			0.3	0.5
36			0.4	0.2
37			0.3	1.0
38			0.2	
39			0.6	
40			0.1	
41			0.6	
42			0.6	
43			0.9	
44			0.4	
45			0.7	
46			0.4	
47			0.3	
48			0.3	
49			0.3	

each of the four networks. The solutions are reported in Table 10, which is organized exactly the same way as Table 9.

For Network A, (WMN1) was able to solve seven instances to optimality and could not close the gap in the remaining instance. On the other hand, with the addition of valid inequalities, each of the eight instances was solved to optimality. Furthermore, the valid inequalities improved the solution time in seven out of eight instances.

For Network B, two instances were solved to optimality with (WMN1) whereas the addition of valid inequalities successfully closed the optimality gap on all four instances. In all of these instances, the valid inequalities improved the solution times.

Unfortunately, none of the instances of Network C could be solved to optimality with either of the two models. The valid inequalities improved the best feasible solution in three out of eight instances. In two of the remaining five instances, where both alternatives returned the same best feasible solutions, the valid inequalities improved the upper bound and, consequently, the optimality gap. In the remaining three instances, the valid inequalities did not lead to an improvement over the original model. Note, however, that the valid inequalities prevented premature termination of CPLEX run due to memory problems in two instances.

For Network D, (WMN1) computed a better solution in one instance and the model with valid inequalities returned a better solution in another instance. In the remaining four instances, where both alternatives returned the same best feasible solutions, the valid inequalities improved the upper bound and the optimality gap in only one instance.

Note that the best feasible solution obtained using the valid inequalities was at least as good as that obtained from the original model on 25 out of the 26 instances. Moreover, these valid inequalities usually improved the solution times and decreased the memory requirements.[3] Therefore, we decided to incorporate these valid inequalities into (WMN1) in our subsequent experiments. Henceforth, this optimization model will be denoted by
(WMN1′).

5.3. Three-stage heuristic method

In this section, we present our results obtained by using the three-stage heuristic, denoted by (3S), outlined in Section 4 on each of the four networks.

We report our results for Networks A and B in Table 11, which is comprised of three sets of columns. The first set of columns reports the details of the instance. The second set of columns presents the optimal solution, and the CPU time of (WMN1′). The last set of columns is devoted to the three-stage heuristic and presents the service level of the heuristic solution, the gap of the heuristic solution, the upper bound (UB) given by the optimal value of the first stage model (WMN-S1), CPU time of Stage 1, and the total CPU time. The gap of the heuristic solution is calculated as the difference between the optimal solution and the heuristic solution divided by the latter.

(WMN1′) could solve all instances of Network A to optimality within 20 min of CPU time. The three-stage heuristic approach, on the other hand, took less than 4 min on each of the eight instances. Note that the computational effort for each of the second and the third stages was quite negligible. On five instances (indicated by boldface entries in Table 11), the solution computed by the heuristic approach matched the optimal solution of (WMN1′). On the remaining three instances, the service level of the optimal solution was at most 7% higher than the service level of the heuristic solution. Therefore, the performance of (3S) on this network was quite satisfactory.

(WMN1′) solved all instances of Network B to optimality, but the solution time varied from 5 min to 42 min on these instances. The three-stage heuristic approach, on the other hand, took less than 17 s on each of the four instances. The heuristic solutions matched the optimal solutions in two instances (indicated by boldface entries in Table 11) and the service level of the optimal solution was at most 6% higher than the service level of the heuristic solution on the remaining instances.

Table 11 also reveals that the upper bounds obtained from the optimal value of (WMN-S1) are indeed quite tight on all instances of Network A and Network B. For each network, the upper bound is tighter for larger values of T. It follows that, on each of these instances, (WMN-S1) yields a reasonably good upper bound with relatively little computational effort.

[3] We observe similar improvements when the valid inequalities (44) are incorporated into (WMN0) supplemented with (30). However, the performance of the resulting model still remains inferior to that of (WMN1) in all of the instances, even without adding valid inequalities (44). Therefore, we do not report these experiment results.

Table 8
Outcomes of (AUX) and Algorithm 1.

Network			(AUX)		Algorithm 1						
Label	$\|N\|$	$\|E\|$	Optimal value	Time	Time	$\|\mathcal{S}\|$	$\|\mathcal{S}_1\|$	$\|\mathcal{S}_2\|$	$\|\mathcal{S}_3\|$	$\|\mathcal{S}_4\|$	$\|\mathcal{S}_5\|$
A	30	98	2	3	1	242	98	144	–	–	–
B	27	162	4	4	3	1184	162	641	355	26	–
C	49	168	3	11	9	2716	168	2436	112	–	–
D	37	254	5	28	20	14982	254	3281	7650	3652	145

Table 9
Model comparisons.

			(WMN0) with valid inequality (30)				(WMN1)			
Network	G	T	Best feasible	Best bound	Gap (%)	Time	Best feasible	Best bound	Gap (%)	Time
A	3	64	0.9375	1.9386	106.78	72000	**1.0135**	**1.0135**	**0.00**	**27321**
		128	0.9375	2.0590	119.63	72000	**1.0938**	**1.0938**	**0.00**	**2161**
A	4	64	1.1250	2.0352	80.91	72000	**1.2500**	**1.2500**	**0.00**	**565**
		128	1.2500	2.7336	118.69	72000	**1.3125**	**1.3125**	**0.00**	**248**
A	5	64	1.4063	3.2268	129.46	72000	**1.5000**	**1.5000**	**0.00**	**1142**
		128	1.4803	3.5228	137.98	72000	**1.5938**	**1.6071**	**0.84**	72000
A	6	64	1.5000	3.8569	157.13	72000	**1.7500**	**1.7500**	**0.00**	**4186**
		128	1.5000	3.8668	157.79	72000	**1.8750**	**1.8750**	**0.00**	**698**
B	2	64	1.5000	1.8462	23.08	72000	**1.6071**	**1.6071**	**0.00**	**15051**
		128	1.5789	2.9464	86.61	72000	**1.6723**	**1.6723**	**0.00**	**4342**
B	3	64	1.8750	3.7855	101.89	72000	**2.4265**	**2.5000**	**3.03**	72000
		128	1.8750	5.7340	205.82	72000	**2.6250**	**2.6471**	**0.84**	72000
C	4	64	0.7955	4.0320	406.88	72000	**1.1538**	**1.2416**	**7.60**	72000
		128	0.9375	4.3076	359.48	72000	**1.2054**	**1.2500**	**3.70**	72000
C	5	64	0.9375	4.5174	381.85	72000	**1.2500**	**1.6705**	**33.64**	72000
		128	0.9375	4.8660	419.04	72000	**1.2500**	**1.6427**	**31.42**	72000
C	6	64	1.0714	5.2941	394.12	72000	**1.2500**	**2.6032**	**108.25**	72000
		128	0.9375	5.6803	505.90	72000	**1.4063**	**2.7273**	**93.94**	72000
C	7	64	1.0714	5.9186	452.40	72000	**1.2500**	**3.5154**	**181.23**	57087*
		128	1.1250	6.5868	485.49	72000	**1.5625**	**3.7804**	**141.95**	62227*
D	3	64	0.7721	3.9682	413.97	72000	**1.1719**	**1.2629**	**7.77**	72000
		128	0.8750	4.5925	424.85	72000	**1.2500**	**1.3000**	**4.00**	72000
D	4	64	0.9375	4.3766	366.84	72000	**1.5000**	**1.5694**	**4.62**	72000
		128	0.9375	4.9775	430.93	72000	**1.5625**	**1.5892**	**1.71**	72000
D	5	64	1.2500	4.6229	269.83	72000	**1.8750**	**1.8750**	**0.00**	**66596**
		128	1.1719	7.4584	536.45	72000	**1.8750**	**1.9383**	**3.38**	72000

Table 10
Computational results with and without valid inequalities.

			(WMN1)				(WMN1) with valid inequality (44)			
Network	G	T	Best feasible	Best bound	Gap (%)	Time	Best feasible	Best bound	Gap (%)	Time
A	3	64	1.0135	1.0135	0.00	27321	1.0135	1.0135	0.00	**54**
		128	1.0938	1.0938	0.00	2161	1.0938	1.0938	0.00	**48**
A	4	64	1.2500	1.2500	0.00	565	1.2500	1.2500	0.00	**290**
		128	1.3125	1.3125	0.00	248	1.3125	1.3125	0.00	**181**
A	5	64	1.5000	1.5000	0.00	1142	1.5000	1.5000	0.00	**888**
		128	1.5938	1.6071	0.84	72000	1.5938	**1.5938**	**0.00**	**960**
A	6	64	1.7500	1.7500	0.00	4186	1.7500	1.7500	0.00	**908**
		128	1.8750	1.8750	0.00	**698**	1.8750	1.8750	0.00	1145
B	2	64	1.6071	1.6071	0.00	15051	1.6071	1.6071	0.00	**1933**
		128	1.6723	1.6723	0.00	4342	1.6723	1.6723	0.00	**416**
B	3	64	2.4265	2.5000	3.03	72000	2.4265	**2.4265**	**0.00**	**2525**
		128	2.6250	2.6471	0.84	72000	2.6250	**2.6250**	**0.00**	**329**
C	4	64	1.1538	1.2416	7.60	72000	1.1538	**1.1976**	**3.79**	72000
		128	1.2054	1.2500	3.70	72000	1.2054	**1.2327**	**2.27**	72000
C	5	64	1.2500	**1.6705**	**33.64**	72000	1.2500	1.9169	53.35	72000
		128	1.2500	**1.6427**	**31.42**	72000	1.2500	1.6637	33.10	72000
C	6	64	1.2500	**2.6032**	**108.25**	72000	1.2500	3.2791	162.33	72000
		128	1.4063	**2.7273**	**93.94**	72000	**1.5625**	3.0744	96.76	72000
C	7	64	1.2500	**3.5154**	181.23	57087*	**1.6667**	3.6123	**116.74**	72000
		128	1.5625	3.7804	141.95	62227*	**1.8750**	**3.4742**	**85.29**	72000
D	3	64	1.1719	1.2629	7.77	72000	**1.1786**	**1.2234**	**3.80**	72000
		128	1.2500	1.3000	4.00	72000	1.2500	**1.2710**	**1.68**	72000
D	4	64	1.5000	**1.5694**	**4.62**	72000	1.5000	1.5711	4.74	72000
		128	1.5625	**1.5892**	**1.71**	72000	1.5625	1.6088	2.96	72000
D	5	64	**1.8750**	1.8750	0.00	66596	1.7647	2.1281	20.59	72000
		128	1.8750	**1.9383**	**3.38**	72000	1.8750	1.9709	5.11	72000

Table 11
Computational results on Networks A and B.

Network	G	T	(WMN1′) Solution	Time	(3S) Solution	Gap (%)	UB	Time (WMN-S1)	Time (Total)
A	3	64	1.0135	54	0.9868	2.71	1.1707	12	13
		128	1.0938	48	1.0795	1.32			13
A	4	64	1.2500	290	1.1719	6.66	1.3953	50	51
		128	1.3125	181	**1.3125**	0.00			51
A	5	64	1.5000	888	**1.5000**	0.00	1.7021	71	71
		128	1.5938	960	**1.5938**	0.00			71
A	6	64	1.7500	908	**1.7500**	0.00	1.9592	188	188
		128	1.8750	1145	**1.8750**	0.00			188
B	2	64	1.6071	1933	1.5203	5.71	1.7518	5	5
		128	1.6723	416	1.6667	0.34			6
B	3	64	2.4265	2525	**2.4265**	0.00	2.7907	13	17
		128	2.6250	329	**2.6250**	0.00			16

Table 12
Computational results on Networks C and D.

Network	G	T	Best in Table 10 Best feasible	Best bound	(3S) Solution	Gap (%)	UB	Time (WMN-S1)	Time (Total)
C	4	64	1.1538	1.1976	**1.1538**	3.80	1.2917	4092	4094
		128	1.2054	1.2327	1.1932	3.31			4095
C	5	64	1.2500	1.6705	**1.2500**	33.64	1.9355*	7200*	7206
		128	1.2500	1.6427	**1.3393**	22.65			7202
C	6	64	1.2500	2.6032	**1.5000**	73.55	3.0573*	7200*	7204
		128	1.5625	2.7273	**1.6544**	64.85			7205
C	7	64	1.6667	3.5154	**1.6667**	110.92	3.6244*	7200*	7207
		128	1.8750	3.4742	**1.8750**	43.74			7204
D	3	64	1.1786	1.2234	1.1250	8.75	1.3355	953	962
		128	1.2500	1.2710	1.2260	3.67			963
D	4	64	1.5000	1.5694	**1.5000**	4.62	1.6696	4482	4486
		128	1.5625	1.5892	**1.5625**	1.71			4487
D	5	64	1.8750	1.8750	**1.8750**	0.00	1.9955	6533	6544
		128	1.8750	1.9383	**1.8750**	3.38			6544

In Table 12, we report the results of (3S) for Networks C and D. The organization of this table is similar to that of Table 11. Note that we do not have the optimal solution in most of these instances. Therefore, in the second set of columns in Table 12, we report, for each instance, the best feasible solution and the best upper bound presented in Table 10. The gap calculations are also different for the same reason. In the calculation of the gap for the heuristic solution, we substitute the optimal solution with the minimum of the best bound from Table 10 and the upper bound obtained by (WMN-S1). It follows that the computed gap is, in fact, an upper bound on the true gap. Note that (WMN-S1) was solved under a CPU time limit of 2 h.

Under a time limit of 20 h, (WMN1) and (WMN1′) could not solve any instance of Network C to optimality and terminated with large optimality gaps, especially for larger number of gateways. On the other hand, while the first stage model (WMN-S1) was solved to optimality for $G = 4$ within the 2-h time limit, the solver stopped with large optimality gaps for $G \in \{5, 6, 7\}$ due to the time limit. For these instances, given the best feasible solution of the first stage model (WMN-S1) under the time limit, we continued with the second and the third stages of (3S) accordingly. The upper bounds for these instances, which are indicated by asterisks, are the best bounds obtained by (WMN-S1) within the time limit.

The solution computed by our three-stage heuristic (3S) outperformed the best solution computed by (WMN1) and (WMN1′) in 20 h on three out of eight instances of Network C, and matched it on four instances. These instances are indicated by boldface entries in Table 12. In the remaining instance with four gateways and $T = 128$, the heuristic solution was slightly worse than this solution. It is again worth noticing that the second and the third stages could be performed very quickly on each instance.

As for the gaps for Network C, we observe two different cases. For the instances with $G = 4$, the optimality gaps are small for both (WMN1) and (WMN1′), and (WMN-S1) is solved to optimality. Therefore, the upper bounds on the optimal service levels are fairly tight and the computed gaps for the heuristic solution are therefore similar to those of Networks A and B. For the instances with $G \in \{5, 6, 7\}$, however, the solver terminated with fairly large optimality gaps due to the time limit for each of the three models (WMN1), (WMN1′), and (WMN-S1). Consequently, the upper bounds on these instances are rather loose and the computed gaps for the heuristic solutions are therefore fairly large. In an attempt to obtain more accurate gap values, we solved (WMN-S1) without any time limit and obtained the upper bounds 1.4861, 1.7844, and 2.0021 for the instances with $G = 5$, $G = 6$, and $G = 7$, respectively. The corresponding CPU times were 11,571, 39,142, and 132,155 s. Note that the resulting upper bounds from the optimal values of (WMN-S1) are considerably tighter than the ones presented in Table 12. We report the more accurate gaps for the same heuristic solutions using the tighter upper bounds in Table 13.

Table 13 reveals that the optimal solutions on these instances were at most around 20% larger than the heuristic solutions on each of these instances. We again remark that these gaps constitute an upper bound on the true gaps as we employ the upper bounds for the optimal solutions instead of the optimal solutions in their calculation.

The best bounds obtained from Table 10 on the instances of Network D were fairly tight. Therefore, (WMN-S1) could not im-

Table 13
More accurate gaps for the heuristic solutions on Network C.

		(3S)		
G	T	Solution	UB	Gap (%)
5	64	1.2500	1.4861	18.89
	128	1.3393		10.96
6	64	1.5000	1.7844	18.96
	128	1.6544		7.86
7	64	1.6667	2.0021	20.12
	128	1.8750		6.78

prove the upper bounds. However, even though (3S) was completed in less than 2 h on all six instances, it computed solutions that matched the best solution in four of those instances indicated by boldface entries. On this network, the computed gaps of the heuristic solution are below 9% for the instances with $T = 64$, and below 4% for the instances with $T = 128$.

Considering the instances for which we have an optimal solution, Tables 11 and 12 reveal that the gaps of the heuristic solution are below 7%. We believe that not having an optimal solution is mainly responsible for the larger computed gaps presented for the other instances.

5.3.1. Comparison with the heuristic method in [12]

In our earlier paper [12], we proposed a two-stage heuristic method. In this section, we compare the performance of the heuristic approach (3S) with the heuristic proposed in [12], henceforth referred to as (2S).

Similar to (3S) presented in this work, gateway locations are determined in the first stage of (2S). However, a different approach is adopted for the selection of gateway nodes. In the first stage of (2S), wireless links are viewed as wired links of capacity c. Next, gateway nodes are selected so as to minimize the total traffic flow on the wired links of the network under the assumption that the demand of each node is satisfied (i.e., service level is equal to one). For this purpose, the following optimization model, denoted by (G1), is employed:

$$\min \sum_{(i,j)\in E} f_{ij}$$

subject to (7), (8), (10), (11), (18), (19), (21), (22), (24), and

$$f_{ij} \le c z_{ij}, \quad (i,j) \in E,$$

$$\sum_{(h,i)\in E} f_{hi} + \sum_{(i,j)\in E} f_{ij} \le c, \quad i \in \{1,\dots,N\},$$

$$w = 1.$$

In the second stage of (2S), the gateway locations obtained from (G1) are fixed and the optimization model (WMN0) with the valid inequality (30) is solved accordingly.

In Table 14, we compare the heuristic approaches (2S) and (3S) in terms of the solution quality and computation time. This table is organized as follows: The first set of columns reports the details of the instance. In the second and the third set of columns, we present the heuristic solutions and total solution times in seconds obtained from the heuristic methods (2S) and (3S), respectively. We also present the percentage improvements of our new heuristic (3S) over the previous heuristic (2S) in the third set of columns, which is computed by the difference between the service levels of the solutions obtained from (3S) and (2S) divided by the latter. Note that the second stage of (2S) and the first stage of (3S) are both solved under a CPU time limit of 2 h. If the solution of either one is stopped due to the time limit, we indicate that by an asterisk.

Table 14 reveals that (3S) computed a better solution than the previous heuristic (2S) in all twenty-six instances. We remark that

Table 14
Comparison of (2S) and (3S).

			(2S)		(3S)		
Network	G	T	Solution	Time	Solution	Time	Improvement (%)
A	3	64	0.9122	7217*	**0.9868**	13	8.18
		128	0.9375	7217*	**1.0795**	13	15.15
A	4	64	1.0714	7226*	**1.1719**	51	9.38
		128	1.1250	7226*	**1.3125**	51	16.67
A	5	64	1.4063	2409	**1.5000**	71	6.66
		128	1.4583	7263*	**1.5938**	71	9.29
A	6	64	1.5000	7258*	**1.7500**	188	16.67
		128	1.6875	4243	**1.8750**	188	11.11
B	2	64	1.5000	7205*	**1.5203**	5	1.35
		128	1.5000	7205*	**1.6667**	6	11.11
B	3	64	1.8750	7210*	**2.4265**	17	29.41
		128	1.8750	7210*	**2.6250**	16	40.00
C	4	64	1.0417	8163*	**1.1538**	4094	10.76
		128	1.0714	8163*	**1.1932**	4095	11.37
C	5	64	1.1538	7972*	**1.2500**	7206*	8.34
		128	1.2500	7972*	**1.3393**	7202*	7.14
C	6	64	1.2500	7733*	**1.5000**	7204*	20.00
		128	1.5625	7733*	**1.6544**	7205*	5.88
C	7	64	1.2500	8736*	**1.6667**	7207*	33.34
		128	1.5625	8736*	**1.8750**	7204*	20.00
D	3	64	1.0714	7231*	**1.1250**	962	5.00
		128	1.0227	7231*	**1.2260**	963	19.88
D	4	64	1.1250	7287*	**1.5000**	4486	33.33
		128	1.0417	7287*	**1.5625**	4487	50.00
D	5	64	1.4063	7264*	**1.8750**	6544	33.33
		128	1.1458	7264*	**1.8750**	6544	63.64

the solutions computed by (3S) are up to 64% better than the solutions obtained by (2S). Furthermore, the total computational effort required by (3S) was always smaller than that for (2S). These computational results illustrate that our heuristic approach (3S) consistently outperforms the heuristic (2S) proposed in [12].

5.3.2. k-opt hill climbing

Based on our computational experiments and discussions in Section 5.3, the first stage model (WMN-S1) can be solved fairly quickly on smaller instances such as Networks A and B and provides a reasonably good upper bound on the optimal service level. On the other hand, for larger instances such as Networks C and D, even solving (WMN-S1) may require a significant computational effort.

In this section, we consider using the 1-opt and 2-opt hill climbing methods as substitutes for directly solving the first stage model (WMN-S1) on Networks C and D. More precisely, for each number of gateways, we randomly select 10 different initial gateway sets. Adding the constraint (51) to (WMN-S1) with $k = 1$ or $k = 2$, we start our 1-opt or 2-opt hill climbing methods, respectively, from those particular sets of gateways. Note that the constraint (51) ensures that, at each iteration, the next set of gateways can differ from the current set in at most k nodes. Once these methods terminate (i.e., the set of gateway nodes remains the same in two consecutive iterations), we fix the corresponding gateway locations and continue with the second and the third stages of our heuristic approach.

We report our computational results for Networks C and D in Table 15, which is organized as follows: The first set of columns reports the details of the instance. In the second set of columns, we present the heuristic solutions and solution times obtained from (3S) employing (WMN-S1) in the first stage. Here, we again impose a 2-h time limit for solving the first stage model (WMN-S1) and asterisks indicate the instances where (WMN-S1) was not solved to optimality. In the third set of columns, under the title '0-opt', we report the best, the worst, and the average service levels for each instance obtained from (3S) skipping the first stage and di-

Table 15
(WMN-S1) versus *k*-opt comparisons.

Network	G	T	With (WMN-S1)		With (0-opt)			With (1-opt)				With (2-opt)			
			Solution	Time	Best	Worst	Average	Best	Worst	Average	Time	Best	Worst	Average	Time
C	4	64	1.1538	4094	0.9375	0.6048	0.7885	1.1538	1.0227	1.0651	**616**	1.0714	1.0500	1.0693	7403
		128	1.1932	4095	1.0417	0.6667	0.8615	1.1932	1.1250	1.1516	**625**	1.1932	1.1250	1.1646	7406
C	5	64	1.2500	7206*	1.0714	0.6250	0.8773	1.2500	1.2500	1.2500	**657**	1.2500	1.2500	1.2500	12421
		128	1.3393	7202*	1.1719	0.6888	0.9543	1.3393	1.2891	1.3174	**660**	1.3393	1.3281	1.3343	12431
C	6	64	1.5000	7204*	1.2500	0.8036	1.0319	1.5000	1.3235	1.3881	**641**	1.5000	1.3393	1.4357	13870
		128	1.6544	7205*	1.3393	0.9375	1.1235	1.6544	1.5625	1.5809	**639**	1.6544	1.5625	1.6176	13856
C	7	64	1.6667	7207*	1.3393	0.9375	1.1153	**1.7045**	1.5625	1.6579	**778**	**1.7045**	1.6667	1.6818	11095
		128	1.8750	7204*	1.5625	1.0227	1.2135	1.8750	1.8750	1.8750	**768**	1.8750	1.8750	1.8750	11054
D	3	64	1.1250	962	1.0227	0.5288	0.7990	**1.1538**	1.0714	1.1258	**824**	**1.1538**	1.1250	1.1423	5719
		128	1.2260	963	1.0938	0.5515	0.8596	**1.2500**	1.1250	1.2154	**828**	**1.2500**	1.2260	1.2404	5718
D	4	64	1.5000	4486	1.2028	0.5474	0.9566	1.5000	1.4423	1.4885	**1046**	1.5000	1.5000	1.5000	19202
		128	1.5625	4487	1.2500	0.5720	1.0164	1.5625	1.5625	1.5625	**1037**	1.5625	1.5625	1.5625	19226
D	5	64	1.8750	6544	1.2500	0.7500	1.1024	1.8750	1.7308	1.8317	**1487**	1.8750	1.8750	1.8750	28762
		128	1.8750	6544	1.3393	0.8333	1.1579	1.8750	1.8750	1.8750	**1489**	1.8750	1.8750	1.8750	28597

rectly employing each one of the ten random gateway sets in the second and the third stages. Since the solution times of the second and the third stages are negligible, we do not report any solution times. In the fourth and the fifth set of columns, for each instance, we report the best, the worst, and the average service levels obtained from (3S) employing 1-opt and 2-opt, respectively, in the first stage along with the total CPU times obtained over the same ten random initial gateway sets. In this study, we do not consider Networks A and B, for which (WMN-S1) can be solved to optimality within a short period of time.

Table 15 reveals that, compared with employing random initial gateway sets directly in the second stage (i.e., 0-opt), using 1-opt or 2-opt local search methods in the first stage yields significant improvements on the service level of the resulting feasible solution. More precisely, the improvements vary from 13% to 50% for best solutions; from 65% to 173% for worst solutions; and from 34% to 70% for average solutions. These results demonstrate the effectiveness of our *k*-opt hill climbing method.

The best solution obtained from (3S) employing 1-opt in the first stage over 10 random initial gateways matched the solution obtained from (3S) employing (WMN-S1) in the first stage in 11 out of 14 instances and was strictly better in the remaining instances indicated by boldface entries. Furthermore, the total CPU time of (3S) employing 1-opt in the first stage is smaller than the CPU time of (3S) employing (WMN-S1) in the first stage in all of these instances. We therefore conclude that using 1-opt from 10 random initial gateway sets in the first stage is usually much faster than solving (WMN-S1) and may return solutions that are of similar or better quality on larger instances.

Considering the 2-opt hill climbing method, note that its best solutions matched the best solutions obtained from the 1-opt approach in thirteen out of fourteen instances and was slightly below in the remaining instance. This observation suggests that increasing the parameter k in the k-opt hill climbing method does not necessarily guarantee a better solution as the local search method may get stuck at a local optimal solution. Note, however, that the average and the worst-case performances of the 2-opt approach are always at least as good as the average and the worst-case performances of the 1-opt approach, respectively, which suggests that a larger value of k may have the potential to yield better solutions on average. As expected, this increased performance comes at the expense of additional computational effort. Our experiments reveal that the additional time required by the 2-opt approach does not seem to be justified by the quality of the resulting solution on this data set. We therefore recommend using the 1-opt approach as a competitive alternative to directly solving (WMN-S1), especially for larger instances.

5.4. Discussion

These computational experiments indicate that our optimization model (WMN1′) can be used to obtain an exact solution for smaller networks in a reasonable amount of time. For larger networks, our three-stage heuristic approach can compute optimal or near-optimal solutions using considerably lower CPU times. Furthermore, for networks where (WMN-S1) is also difficult to solve, the 1-opt hill climbing method can be used as a stand-alone approach that completely eliminates the need to solve (WMN-S1). Finally, since the second and the third stages of our heuristic approach usually require a fairly low computational effort, for these networks, the 1-opt hill climbing method can be started from several randomly selected initial gateway sets and the best feasible solution among them can be implemented.

6. Conclusions

In this paper, we considered planning and operational problems of WMNs, such as gateway selection, routing, transmission capacity allocation, and power control, employing the physical interference model.

We proposed two mixed integer linear programming formulations for this problem. The first model is adapted from [12] to the setting studied in this paper. The alternative formulation is based on sets of noninterfering links and its size is independent of the number of time slots in a frame. We determined not only the necessary and sufficient conditions in order for a set of links to be activated together in the same time slot but also an appropriate power setting for each transmitting node. Consequently, in the alternative formulation, we completely eliminated the power control and the interference components from the problem. Our computational results revealed that the alternative formulation significantly outperforms the first formulation in terms of the solution quality and computation time.

In the alternative formulation, time slots are allocated to sets of noninterfering transmissions. The main disadvantage of this formulation is that the set of noninterfering transmissions may get exponentially large with increasing network sizes. In that case, commercial solvers may return large optimality gaps or may terminate prematurely due to insufficient memory. For such large networks, we proposed a three-stage heuristic method. In the first stage, by solving a partially relaxed and simplified version of the alternative optimization model, we determine a set of gateways as well as an upper bound for the service level. In the second stage, we determine a routing tree for each gateway. The last stage determines the allocation of time slots to noninterfering transmission sets to form these routing trees. For even larger networks, for which it may take a long time to obtain gateway locations in the first stage, we proposed a *k*-opt hill climbing method that can be employed as a substitute for solving (WMN-S1) in the first stage of our heuristic. Our computational results illustrate the promising performances of these heuristic approaches. Furthermore, we observe that our new heuristic approach significantly outperforms our earlier heuristic approach proposed in [12].

Note that a solution returned by our exact or heuristic approaches specifies the number of time slots that should be allocated to each noninterfering transmission set in a TDMA frame. However, the specific assignment of each time slot to a noninterfering transmission set has not been addressed in our paper since this assignment problem has no effect on the service level. On the other hand, the order in which each time slot is assigned to a noninterfering transmission set may have an impact on other quality of service measures such as the total delay of packages in the network. We intend to work in this direction in the near future.

Acknowledgment

We gratefully acknowledge the insightful comments and suggestions of two anonymous reviewers and the editors, which led to several improvements in our manuscript.

References

[1] Akyildiz IF, Wang X, Wang W. Wireless mesh networks: a survey. Comput Networks 2005;47(4):445–87.
[2] Amaldi E, Capone A, Cesana M, Filippini I, Malucelli F. Optimization models and methods for planning wireless mesh networks. Comput Networks 2008;52(11):2159–71.
[3] Aoun B, Boutaba R, Iraqi Y, Kenward G. Gateway placement optimization in wireless mesh networks with QoS constraints. IEEE J Selected Areas Commun 2006;24(11):2127–36.
[4] Badia L, Botta A, Lenzini L. A genetic approach to joint routing and link scheduling for wireless mesh networks. Ad Hoc Networks 2009;7(4):654–64.
[5] Bejerano Y. Efficient integration of multihop wireless and wired networks with QoS constraints. IEEE/ACM Trans Networking 2004;12(6):1064–78.
[6] Benyamina D, Hafid A, Gendreau M. Wireless mesh networks design—a survey. IEEE Commun Surveys Tutor 2012;14(2):299–310.
[7] Capone A, Carello G, Filippini I, Gualandi S, Malucelli F. Routing, scheduling and channel assignment in wireless mesh networks: optimization models and algorithms. Ad Hoc Networks 2010;8:545–63.
[8] Chandra R, Qiu L, Jain K, Mahdiant M. Optimizing the placement of internet TAPs in wireless neighborhood networks. In: Proceedings of the 12th IEEE international conference on network protocols; 2004.
[9] Croes GA. A method for solving traveling-salesman problems. Oper Res 1958;6(6):791–812.
[10] Erceg V, Greenstein LJ, Tjandra SY, Parkoff SR, Gupta A, Kulic B, et al. An empirically based path loss model for wireless channels in suburban environments. IEEE J Selected Areas Commun 1999;17(7):1205–11.
[11] Fischetti M, Lodi A. Local branching. Math Program 2003;98(1–3):23–47.
[12] Gokbayrak K, Yıldırım EA. Joint gateway selection, transmission slot assignment, routing and power control for wireless mesh networks. Comput Oper Res 2013;40(7):1671–9.
[13] Gupta P, Kumar PR. The capacity of wireless networks. IEEE Trans Inf Theory 2000;46(2):388–404.
[14] Karnik A, Iyer A, Rosenberg C. Throughput-optimal configuration of fixed wireless networks. IEEE/ACM Trans Networking 2008;16(5):1161–74.
[15] Lin S. Computer solutions of the traveling salesman problem. Bell Syst Tech J 1965;44(10):2245–69.
[16] Luo J, Rosenberg C, Girard A. Engineering wireless mesh networks: joint scheduling, routing, power control, and rate adaptation. IEEE/ACM Trans Networking October 2010;18(5):1387–400.
[17] Papadaki K, Friderikos V. Approximate dynamic programming for link scheduling in wireless mesh networks. Comput Oper Res 2008;35(12):3848–59.
[18] Papadimitriou CH, Steiglitz K. Combinatorial optimization: algorithms and complexity. Courier Corporation; 1982.
[19] Papanastasiou S, Mittag J, Strm EG, Hartenstein H. Bridging the gap between physical layer emulation and network simulation. In: IEEE WCNC 2010 Proceedings; 2010.
[20] Pathak PH, Dutta R. A survey of network design problems and joint design approaches in wireless mesh networks. IEEE Commun Surv Tutor 2011;13(3):396–428.
[21] Quintas D, Friderikos V. Energy efficient spatial TDMA scheduling in wireless networks. Comput Oper Res 2012;39(9):2091–9.
[22] Targon V, Sanso B, Capone A. The joint gateway placement and spatial reuse problem in wireless mesh networks. Comput Networks 2010;54(2):231–40.
[23] Yang L, Chengguo L, Miao Z, Yuan W, Yang L. Radio wave propagation path loss in the irregular terrain environments. In: 2009 3rd IEEE international symposium on microwave, antenna, propagation and EMC technologies for wireless communications. IEEE; 2009. p. 627–30.