Optimizing link rate assignment and transmission scheduling in WMN through compatible set generation

Yuan Li, Michal Pioro, Di Yuan and Jinshu Su

Linköping University Post Print

N.B.: When citing this work, cite the original article.

The original publication is available at www.springerlink.com:

Yuan Li, Michal Pioro, Di Yuan and Jinshu Su, Optimizing link rate assignment and transmission scheduling in WMN through compatible set generation, 2016, Telecommunications Systems, (61), 2, 325-335.

<http://dx.doi.org/10.1007/s11235-015-0004-y>

Copyright: Springer Verlag (Germany)

<http://www.springerlink.com/?MUD=MP>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-125822>

Optimizing Link Rate Assignment and Transmission Scheduling in WMN through Compatible Set Generation

Yuan Li · Michał Pióro · Di Yuan · Jinshu Su

Received: date / Accepted: date

Abstract Radio links in wireless mesh networks (WMN) can select one of several modulation and coding schemes (MCS). A MCS assignment influences links data rates and their mutual interference, and therefore should be optimized. We consider joint optimization of link rate assignment and transmission scheduling in order to maximize the minimal flow in a WMN. One of the main difficulties stems from the requirement that each link has to use only one selected MCS for all its transmissions. This requirement leads to a complicated exact branch-and-price method, which is quite time-consuming for networks of practical size. Thus, we propose an original heuristic based on simulated annealing that utilizes specific characteristics of the problem. The method provides a balance between sub-optimality of the obtained solutions and the running time. The presented method is the main purpose and novelty of the paper. An extensive numerical study illustrates effectiveness of the proposed approach.

Y. Li,
Department of Electrical and Information Technology, Lund
University, Lund, Sweden
E-mail: yuan.li@eit.lth.se

M. Pióro,
Department of Electrical and Information Technology, Lund
University, Lund, Sweden
Institute of Telecommunications, Warsaw University of Tech-
nology, Warsaw, Poland
E-mail: michal.pioro@eit.lth.se

D. Yuan,
Department of Science and Technology, Linköping University,
Linköping, Sweden
E-mail: diyua@itn.liu.se

J. Su
School of Computer, National University of Defense Technol-
ogy, Changsha, China
E-mail: sjs@nudt.edu.cn

Keywords static link rate assignment · branch-and-price · simulated annealing

1 Introduction

Wireless Mesh Networks (WMN) is a well-established networking paradigm for residential and metropolitan areas [1]. Such key WMN features as high network capacity, low cost, self-healing and self-organization make them easy to deploy and efficient in many applications such as broadband wireless, neighborhood and enterprise networking, transportation system, etc. WMN networking has attracted extensive research on such aspects as topology design, routing optimization, power control, channel assignment, and transmission scheduling for maximizing the traffic throughput, taking the fairness aspect into account [1–3]. The presented paper considers a problem of joint optimization of link rate assignment and link transmission scheduling under the TDMA-based MAC protocol assumption. A link transmission schedule is devised so that active links do not interfere with each other at any time instant, and each radio link uses a fixed modulation and coding scheme (MCS) when it is active. The objective is to maximize the minimal bandwidth assigned to the network routes, improving traffic fairness and throughput.

The most common MAC protocols used in wireless networks are based on CSMA/CA and TDMA. From the traffic throughput viewpoint, CSMA/CA is not well suited for multihop wireless networks (as WMN) because of its conservative medium access and the hidden terminal problem which causes collisions at a receiver for transmitters which cannot hear each other. TDMA can avoid such collisions since each transmitter works in its dedicated time slots and the transmission scheduling

can be arranged in a way avoiding interference of the signals received at the nodes. TDMA also brings a better spatial reuse that allows more links to be simultaneously active [2, 4]. At the same time, WMN supports several MCS allowing for radio transmissions at different data rates. Assigning MCS with a smaller data rate to radio links can potentially make more links active simultaneously to improve the traffic throughput.

From the modeling standpoint, we incorporate directional antennas into the optimization model, extending the work of [6], since this can considerably improve traffic throughput [7, 8]. The presented model is non-compact and can be solved exactly only through branch-and-price (B&P) – a standard integer programming approach applied in many optimization problems of wireless networks [11]. In our problem we need to generate the compatible set (a set of simultaneously active links) in B&P [5]. Unfortunately, B&P does not work efficiently for large networks in the considered problem setting due to the static way of using MCS. When links use MCSs in a dynamic way, the problem becomes simpler (but not much). Therefore, we propose a simulated annealing heuristic. Our approach is developed using an exact interference model, a SINR model of [12], and thus is different from other heuristics based on non-exact interference models, e.g., a conflict graph, see [13], [14]. That is, we firstly build the model using appropriate SINR constraints, and then apply our heuristic.

In the paper we present a centralized algorithm for joint optimization of static link rate assignment and transmission scheduling that can be used for finding performance bounds on traffic throughput in WMN. We summarize the main contribution and novelty of this paper as follows.

- The considered problem is formulated as a mixed-integer program which can be solved exactly by a B&P algorithm.
- A novel, efficient and easy to implement heuristic algorithm is proposed. The heuristic makes use of specific problem characteristics.
- An extensive numerical study is presented to illustrate effectiveness of the proposed heuristic in comparison with the exact algorithm.

The paper is organized as follows. Section 2 summarizes the related work and explains our motivation. The network model and the basic optimization problem are described in Section 3. Section 4 presents an exact (branch-and-price) method. Then, in Section 5 a simulated annealing heuristic for the considered problem is introduced. Numerical results comparing the exact method and the heuristic are presented in Section 6. Finally, Section 7 concludes the the paper.

2 Related work

Fairness in carrying traffic flows in WMN is an important research topic which has attracted a considerable attention, see [15, 16] and references therein. It turns out, rate control mechanism can improve the fairness in question. In [17], the authors point out the trade-off between link rate assignment and spatial reuse, and investigate the benefits of link rate selection on the max-min flow in WMN. Paper [18] studies a weighted max-min flow rate fairness problem by jointly optimizing link rate adaptation and power control. Various physical layer link rate control functions are investigated. Paper [6] presents mixed-integer programming models for max-min flow optimization with link rate selection and transmission scheduling. The rate control methods are divided to two categories: (i) dynamic rate adaptation where each router can regulate the transmission rate adapted to the changing channel condition; (ii) static rate assignment assuming that each link uses a fixed rate for all its transmissions.

Most previous work for joint optimization of transmission scheduling and link rate control assumes dynamic rate adaptation. A relevant mixed-integer programming model is given in [19]. Models and algorithms for joint optimization of rate adaptation with power control, channel assignment and routing are presented in [5, 20, 21].

Although link rate adaptation through dynamic use of MCSs is potentially an efficient way to react to changing radio channel conditions [22], in our opinion static assignment of MCSs to links is still of primary importance. First, as pointed out in [23], for some cases static MCS assignment is able to perform better than the adaptive one since the latter relies on how well the channel can be detected. Also, the hardware complexity of routers is reduced for static MCS assignment. Second, the presented model is used at the design stage, and adaptive rate selection is not of direct interest as only average channel condition is assumed. Also, at the design stage, the dynamic use of MCS reflects something different from the adaptiveness considered in [22], namely a preplanned schedule of changing the MCS of a link from an activation period to an activation period—a solution that may not be practical.

3 Problem description

3.1 Network model

The considered WMN is modeled as a directed graph $(\mathcal{V}, \mathcal{E})$. The set of nodes $\mathcal{V}$ is composed of a subset of gateways $\mathcal{G}$ and a subset of mesh routers $\mathcal{R}$. The nodes

Table 1: MCSs according to IEEE 802.11a.

MCS m	SINR threshold $\hat{\gamma}^m$	raw bitrate B^m
BPSK 1/2	3.5 dB	6 Mbps
BPSK 3/4	6.5 dB	9 Mbps
QPSK 1/2	6.6 dB	12 Mbps
QPSK 3/4	9.5 dB	18 Mbps
16-QAM 1/2	12.8 dB	24 Mbps
16-QAM 3/4	16.2 dB	36 Mbps
64-QAM 2/3	20.3 dB	48 Mbps
64-QAM 3/4	22.1 dB	54 Mbps

are connected by directed radio links forming the set $\mathcal{E}$. For each link (arc) $e \in \mathcal{E}$, its head node is denoted with $a(e) \in \mathcal{V}$ while its tail node by $b(e) \in \mathcal{V}$, both equipped with directional antennas.

Let $\mathcal{M}$ be the set of modulation and coding schemes. When a link e is active, its data rate depends on the selected MCS $m \in \mathcal{M}$. For an IEEE 802.11 based WMN, available MCSs are listed in Table 1 [17]. In the table, each MCS corresponds to a SINR threshold $\hat{\gamma}^m$ and a raw bitrate B^m . The SINR threshold ($\hat{\gamma}^m$) is expressed in the logarithmic scale (in dB) and can be converted to the linear scale (γ^m) through equality $\gamma^m = 10^{\frac{\hat{\gamma}^m}{10}}$.

Let $\mathcal{A}$ be a set of links which can be active simultaneously. For each active link $e \in \mathcal{A}$, the *signal-to-interference-and-noise ratio* (SINR) condition (1) should be satisfied, i.e., the power received at the tail node $b(e)$ from the head node $a(e)$ divides by the sum of the noise power η and the interference from the active links other than link e should be greater than or equal to a certain SINR threshold. The value of the SINR threshold γ^m depends on the MCS m used by link e . Note that $P_e^{e'}$ denotes the power received at $b(e)$ from $a(e')$.

$$\frac{P_e^e}{\eta + \sum_{e' \in \mathcal{A} \setminus \{e\}} P_e^{e'}} \geq \gamma^m, e \in \mathcal{A}. \quad (1)$$

For each node, it can either be inactive or transmit, or receive. This is formulated as (2) where $\delta^+(v)$ represents the set of links outgoing from node v , and $\delta^-(v)$ denotes the set of links incoming to node v . Equipped with several directional antennas, a node can either transmit to or receive from several other nodes, which is different from the situation when only omni-directional antennas are deployed.

$$\mathcal{A} \cap \delta^+(v) \neq \emptyset \Rightarrow \mathcal{A} \cap \delta^-(v) = \emptyset, v \in \mathcal{V} \quad (2a)$$

$$\mathcal{A} \cap \delta^-(v) \neq \emptyset \Rightarrow \mathcal{A} \cap \delta^+(v) = \emptyset, v \in \mathcal{V}. \quad (2b)$$

In the considered problem, a route for each mesh router $r \in \mathcal{R}$ is specified. Such a route, R_r , is a directed path from a selected gateway $g \in \mathcal{G}$ to the mesh router r . We consider only the routes for the downstream data

but the model can be extended to the upstream routes as well. The load of link $e \in \mathcal{E}$ during the network operation time T should not exceed its capacity c_e , which is formulated in (3). In (3), $\mathcal{Q}_e$ is the set of routes traversing link e , and f_r is the volume of data sent on route $R_r, r \in \mathcal{R}$ during time T .

$$\sum_{r \in \mathcal{Q}_e} f_r \leq c_e, e \in \mathcal{E}. \quad (3)$$

We will use the notion of *compatible set* (see [5,6]) to represent a set of simultaneously active links that does not violate constraints (1) and (2) subsequently. In a compatible set $\mathcal{C}$, each link is associated with a MCS $m_e(\mathcal{C}), e \in \mathcal{C}$. Given a list of compatible set $\mathcal{C}_i, i \in \mathcal{I}$ ($\mathcal{I} = \{1, 2, \dots, I\}$) and its corresponding activity time interval t_i , the capacity of link e is defined as $c_e = \sum_{i \in \mathcal{I}} B_{ei} t_i$. The notation B_{ei} represents the data rate of link e in compatible set $\mathcal{C}_i$. If link e is in compatible set $\mathcal{C}_i$, then $B_{ei} = B^{m_e}$; otherwise, $B_{ei} = 0$. For the considered static MCS assignment problem, each link uses a MCS m_e whenever it is included in an active compatible set, that is, if $t_i > 0$ then $m_e(\mathcal{C}_i) = m_e$. Note that $\sum_{i \in \mathcal{I}} t_i = T$ and T is the considered network operation time.

The flow on each route can be realized by a proper activation of compatible sets from the list $\mathcal{I}$ in their time intervals t_i , and using the packet relay/delay mechanism along each route.

3.2 Optimization problem

Given a list of compatible sets $\mathcal{C}_i, i \in \mathcal{I}$, the space of feasible capacities for all links $c = (c_e, e \in \mathcal{E})$ with respect to non-negative continuous variables $t = (t_i, i \in \mathcal{I})$ is fully characterized by formulas (1), (2) and (3). Using this characterization, we can formulate many problems with different throughput-related objectives, e.g., maximizing the minimum flow over all routes, minimizing the maximum link load, and several others [6]. We assume the criterion of maximizing the minimal flow; then the corresponding mixed-integer formulation is given in (4). The binary variables U_e^m denote whether link e uses MCS m ($U_e^m = 1$) or not ($U_e^m = 0$).

In objective of (4a), f represents the uniform flow sent over each route $R_r, r \in \mathcal{R}$. Constraint (4b) splits the network operation time T among all the compatible sets $\mathcal{C}_i, i \in \mathcal{I}$. Next, constraint (4c) makes sure that the load of link e does not exceed its capacity $c_e = \sum_{i \in \mathcal{I}} B_{ei} t_i$. The load of link e is the total amount of data sent over link e and equals $n_e f$. The quantity n_e is the number of routes using link e , and each route is assigned flow f . Then, constraint (4d) ensures that each link can use only one MCS. Finally, with a binary

constant r_{ei}^m which is equal to 1 when link e belongs to compatible set $\mathcal{C}_i$ and uses MCS m , constraint (4e) guarantees that for each link, if it belongs to a compatible set but is associated with a MCS different from the one specified by $U_e^m = 1$, then this compatible set will not be active, i.e., the corresponding t_i is forced to be 0.

MaxMinFlow: maximize f (4a)

$$\sum_{i \in \mathcal{I}} t_i = T \quad (4b)$$

$$n_e f \leq \sum_{i \in \mathcal{I}} B_{ei} t_i, e \in \mathcal{E} \quad (4c)$$

$$\sum_{m \in \mathcal{M}} U_e^m = 1, e \in \mathcal{E} \quad (4d)$$

$$\sum_{i \in \mathcal{I}} r_{ei}^m t_i \leq U_e^m T, e \in \mathcal{E}, m \in \mathcal{M} \quad (4e)$$

$$t \geq 0, f - \text{continuous}, U - \text{binary}. \quad (4f)$$

Actually, the difficulty in solving model (4) is that each link uses a fixed MCS over all compatible sets used. If a link could switch MCSs between different compatible sets, the binary variable U and constraints (4d), (4e) could be removed, resulting in a formulation which is a linear program (LP) in variables t, f .

4 Exact optimization: branch-and-price

Certainly, the number of compatible sets grows exponentially with size of the network and hence it is virtually impossible to predefine the list of the compatible sets $\mathcal{C}_i, i \in \mathcal{I}$ to reach the optimum of (4). Thus, the compatible sets should be generated during the solution process. Since the considered formulation (4) contains integer (binary in our case) variables, this has to be done through branch-and-price (B&P) – a combination of branch-and-bound (B&B) and column generation [9].

Each node of the B&P tree represents an LP problem referred to as a *restricted master problem* (RMP). The problem is defined by a list $\mathcal{N} = ((N_u^e, N_1^e, N_0^e), e \in \mathcal{E})$ and denoted by $\text{RMP}(\mathcal{N})$. For each link $e \in \mathcal{E}$, N_u^e, N_1^e, N_0^e are mutually disjoint subsets of $\mathcal{M}$ and $N_u^e \cup N_1^e \cup N_0^e = \mathcal{M}$. Moreover, $|N_1^e| \leq 1$, and if $N_1^e = \{m_e\}$ then $N_0^e = \mathcal{M} \setminus \{m_e\}$ and $N_u^e = \emptyset$. $\text{RMP}(\mathcal{N})$ is formulated as (5).

The entities in the brackets denote the dual variables corresponding to the constraints (5b)–(5e). Clearly, dual variables π_e, β_e^m are non-negative, and α, ζ_e are unrestricted in sign.

Now consider a fixed list $\mathcal{N} = ((N_u^e, N_1^e, N_0^e), e \in \mathcal{E})$ and the corresponding problem $\text{RMP}(\mathcal{N})$ with a given initial list of compatible sets $\mathcal{L} = \{\mathcal{C}_i, i \in \mathcal{I}\}$. Certainly, the considered problem can be simplified by (i) eliminating in (5d) and (5e) those variables U_e^m that are fixed by (5g) and (5h) by their actual values, and (ii) skipping (5g) and (5h). The resulting LP problem is

solved through column (i.e., compatible set) generation.

RMP($\mathcal{N}$): maximize f (5a)

$$\sum_{i \in \mathcal{I}} t_i = T \quad [\alpha] \quad (5b)$$

$$n_e f \leq \sum_{i \in \mathcal{I}} B_{ei} t_i, e \in \mathcal{E} \quad [\pi_e] \quad (5c)$$

$$\sum_{m \in \mathcal{M}} U_e^m = 1, e \in \mathcal{E} \quad [\zeta_e] \quad (5d)$$

$$\sum_{i \in \mathcal{I}} r_{ei}^m t_i \leq U_e^m T, e \in \mathcal{E}, m \in \mathcal{M} \quad [\beta_e^m] \quad (5e)$$

$$U_e^m \geq 0, e \in \mathcal{E}, m \in N_u^e \quad (5f)$$

$$U_e^m = 1, e \in \mathcal{E}, m \in N_1^e \quad (5g)$$

$$U_e^m = 0, e \in \mathcal{E}, m \in N_0^e. \quad (5h)$$

$$t_i \geq 0, i \in \mathcal{I}. \quad (5i)$$

The problem dual to the transformed $\text{RMP}(\mathcal{N})$ is as follows:

DP($\mathcal{N}$): minimize $\alpha T + \sum_{e \in \mathcal{E}} \zeta_e$ (6a)

$$\sum_{e \in \mathcal{E}} \pi_e n_e = 1 \quad (6b)$$

$$\sum_{e \in \mathcal{E}} \pi_e B_{ei} - \sum_{e \in \mathcal{E}} \sum_{m \in \mathcal{M}} \beta_e^m r_{ei}^m \leq \alpha, i \in \mathcal{I} \quad (6c)$$

$$\beta_e^m T \leq \zeta_e, e \in \mathcal{E}, m \in \mathcal{M} \quad (6d)$$

$$\text{all variables } \alpha, \pi, \zeta, \beta \text{ continuous, } \beta, \pi \geq 0. \quad (6e)$$

The following proposition, proved in Appendix A, simplifies generating compatible sets.

Proposition 1 *Let $\alpha^*, \pi^*, \zeta^*, \beta^*$ be an arbitrary optimal solution of the dual problem $\text{DP}(\mathcal{N})$ (6). Then, $\alpha^*, \pi^*, \zeta = 0, \beta = 0$ is also an optimal solution of $\text{DP}(\mathcal{N})$.*

Due to Proposition 1, the procedure for solving problem $\text{RPM}(\mathcal{N})$ through column generation consists of (i) finding a compatible set that maximizes the value of $\sum_{e \in \mathcal{E}} \pi_e^* B_{ei}$ (for given optimal dual π^* of $\text{DP}(\mathcal{N})$), (ii) adding the generated set to the list $\mathcal{L}$ if the maximized value is greater than α^* , and (iii) resolving $\text{RPM}(\mathcal{N})$ and coming back to step (i) if $\mathcal{L}$ has been extended. The pricing problem (PP), i.e., finding a compatible set in step (i), can be formulated as model (7).

In the model (7), y_e^m are binary variables, and if link e is active and uses MCS m , $y_e^m = 1$; otherwise, $y_e^m = 0$. Y_e are binary variables representing whether link e is active ($Y_e = 1$) or not ($Y_e = 0$). Thus the data rate of link e can be computed as $B_e = \sum_{m \in \mathcal{M}} B^m y_e^m$ which is in the objective of (7). When generating a compatible set, if the MCS of link $e \in \mathcal{E}$ is fixed, the link must be either inactive, or active with MCS m_e – this is assured by constraints (7b) and (7c). Constraint (7d) implies that a link can only use one MCS when it is active in a compatible set. Next, constraints (7e) – (7h) define variables X_v^+, X_v^- , indicating whether the node v transmits ($X_v^+ = 1$) or receives ($X_v^- = 1$). Then the half duplex condition is assured by the constraint (7i). Constraints (7j) and (7k) imply that the auxiliary

variables $z_{ee'}^m$ equal to $y_e^m \cdot Y_{e'}$. That is, $z_{ee'}^m = 1$ if link e is active by using MCS m , and link e' is also active, no matter which MCS it uses; otherwise $z_{ee'}^m = 0$. Constraint (7l) together with (7j) and (7k) express the SINR constraint, which is defined in (1).

$$\text{PP: maximize } \sum_{e \in \mathcal{E}} \pi_e^* \sum_{m \in \mathcal{M}} B^m y_e^m \quad (7a)$$

$$y_e^m \leq 1, e \in \mathcal{E}, m \in N_1^e \quad (7b)$$

$$y_e^m = 0, e \in \mathcal{E}, m \in N_0^e \quad (7c)$$

$$Y_e = \sum_{m \in \mathcal{M}} y_e^m, e \in \mathcal{E} \quad (7d)$$

$$Y_e \leq X_v^+, e \in \delta^+(v), v \in \mathcal{V} \quad (7e)$$

$$X_v^+ \leq \sum_{e \in \delta^+(v)} Y_e, X_v^+ \leq 1, v \in \mathcal{V} \quad (7f)$$

$$Y_e \leq X_v^-, e \in \delta^-(v), v \in \mathcal{V} \quad (7g)$$

$$X_v^- \leq \sum_{e \in \delta^-(v)} Y_e, X_v^- \leq 1, v \in \mathcal{V} \quad (7h)$$

$$X_v^+ + X_v^- \leq 1, v \in \mathcal{V} \quad (7i)$$

$$z_{ee'}^m \geq y_e^m + Y_{e'} - 1, e, e' \in \mathcal{E}, m \in \mathcal{M} \quad (7j)$$

$$z_{ee'}^m \leq y_e^m, z_{ee'}^m \leq Y_{e'}, e, e' \in \mathcal{E}, m \in \mathcal{M} \quad (7k)$$

$$\eta y_e^m + \sum_{e' \in \mathcal{E} \setminus \{e\}} P_{e'}^e z_{ee'}^m \leq \frac{1}{\gamma^m} P_e^e y_e^m, \quad e \in \mathcal{E}, m \in \mathcal{M} \quad (7l)$$

$$y, Y - \text{binary}, X, z - \text{non-negative continuous}. \quad (7m)$$

Clearly, a compatible set defined as $\mathcal{C} = \{e \in \mathcal{E} : Y_e = 1\}$ can be obtained by solving model (7), where for each active link $e \in \mathcal{C}$, the used MCS m_e is specified by the MCS m for which $y_e^m = 1$.

5 Heuristic method

The exact B&P method described above is not straightforward to implement and, what is more important, can be excessively time consuming even for medium-size networks of, say 10-20 nodes. Therefore, heuristic methods become important. In this section we describe a heuristic method based on simulated annealing (SA, see [24], and the list of literature there).

5.1 SA solution space and neighborhoods

Consider the problem MaxMinFlow (4), suppose that $m^0 = (m_e^0, e \in \mathcal{E})$ is the vector of the MCSs assigned to links in the compatible sets that are used in certain optimal solution of (4). Let $\hat{\mathcal{I}}$ denote the list of all compatible sets of the considered network and define its sublist $\mathcal{I}$ composed of the compatible sets consistent with m^0 , i.e., $\mathcal{I} = \{i \in \hat{\mathcal{I}} : \forall e \in \mathcal{E}, e \in \mathcal{C}_i \Rightarrow r_{ei}^{m_e^0} = 1\}$. Then, clearly, an optimal solution of (4) can be found by the LP problem (8) (formulated in the standard form). This is correct since the list $\mathcal{I}$ of allowable compatible sets in (8) fixes the MCS values for the links so we

Algorithm 1 Transition from $\mathcal{S}$ to $\mathcal{S}'$

Step 1: $w_e := \phi \times \frac{\pi_e^*}{\sum_{e' \in \mathcal{E}} \pi_{e'}^*} + (1 - \phi) \times \frac{2 \times \text{random}(0,1)}{E}$.

$e_{max} := \text{argmax}(w_e)$, $e_{min} = \text{argmin}(w_e)$.

Step 2: Randomly select an MCS m ($m \neq m_{e_{max}}^S$) allowable for e_{max} .

Step 3:

for $i := 1, 2, \dots, E$ **do**

set $p = \text{random}(0, 1)$

if $p \geq \theta$ **then**

$m^{S'}(e_{min}, i) := 0$ (delete e_{min} from $\mathcal{C}_i^S$)

end if

if $p < \theta$ or e_{max} is active **then**

$m^{S'}(e_{max}, i) := m$

(change MCS of e_{max} in $\mathcal{C}_i^S$)

if m not feasible for e_{max} in $\mathcal{C}_i^{S'}$ **then**

$m^{S'}(e_{max}, i) := 0$

end if

end if

end for

Step 4: If e_{min} becomes inactive in all compatible sets of $\mathcal{S}'$, select randomly a compatible set and activate e_{min} with its previous MCS. If this is infeasible, then select randomly another compatible set until e_{min} becomes active or all compatible sets have been checked.

Step 5: If e_{max} becomes inactive in all compatible sets of $\mathcal{S}'$, select randomly a compatible set and activate e_{max} with MCS m . If this is infeasible, then select randomly another compatible set until e_{max} becomes active or all compatible sets have been checked.

Step 6: If either e_{min} or e_{max} cannot be made active, go to Step 1. Otherwise, return $\mathcal{S}'$ and stop.

can skip binary variables U and constraints (4d)–(4e) in (4). LP (8) has $E + 1$ constraints (where $E = |\mathcal{E}|$) and thus there are at most $E + 1$ nonzero variables in any optimal basic solution [10]. Assuming $f^0 > 0$, we conclude that there are at most E nonzero variables t_i ($i \in \mathcal{I}$) in an optimal basic solution of (8), i.e., at most E compatible sets are finally used. Therefore, looking for a list of compatible sets yielding an optimal solution of the problem MaxMinFlow (4), its cardinality can be restricted to E .

$$\text{maximize } f \quad (8a)$$

$$\sum_{i \in \mathcal{I}} t_i = T \quad (8b)$$

$$n_e f + s_e = \sum_{i \in \mathcal{I}} B_{ei} t_i, e \in \mathcal{E} \quad (8c)$$

$$t, s - \text{non-negative continuous}. \quad (8d)$$

The set of all feasible solutions $\mathcal{S}$ of SA is constructed as follows. Let $|\mathcal{I}| = |\mathcal{E}|$. A function $\mathcal{S} : \mathcal{E} \times \mathcal{I} \rightarrow$

$\mathcal{M} \cup \{0\}$ is called an (MCS) assignment if for each $e \in \mathcal{E}$ there exists $m_e^S \in \mathcal{M}$ such that $\mathcal{S}(e, i) \in \{m_e^S, 0\}$ for all $i \in \mathcal{I}$, and m_e^S is allowable for link e . In other words, for each link e , assignment $\mathcal{S}$ fixes the MCS to be used for all $i \in \mathcal{I}$ ($\mathcal{S}(e, i) = m_e^S$), at the same time for some $i \in \mathcal{I}$ allowing not to activate the link at all ($\mathcal{S}(e, i) = 0$). Consequently, an assignment $\mathcal{S}$ determines a sequence of subsets of links $\mathcal{C}^S = (\mathcal{C}_1^S, \mathcal{C}_2^S, \dots, \mathcal{C}_E^S)$ defined by $\mathcal{C}_i^S = \{e \in \mathcal{E} : \mathcal{S}(e, i) \neq 0\}$, $i \in \mathcal{I}$. An assignment $\mathcal{S}$ is called feasible if all sets in the sequence $\mathcal{C}^S$ are compatible sets and each link belongs to at least one of such defined compatible sets. Then all feasible solutions of SA are identified with all feasible assignments $\mathcal{S}$. Note that in any feasible SA solution $\mathcal{S}$, each link e has the same MSC value in all compatible sets of $\mathcal{C}^S$.

The neighborhood of a feasible SA solution $\mathcal{S}$ is composed of all feasible assignments $\mathcal{S}'$ that differ from $\mathcal{S}$ for one or two links. More precisely, $\mathcal{S}'$ is a neighbor of $\mathcal{S}$ if, and only if, $\mathcal{S}'$ is a feasible assignment and one of the subsequent conditions. (i) There exists an link e with $m_e^{S'} = m_e^S$ such that $\mathcal{S}(e, i) \neq 0$, $\mathcal{S}'(e, i) = 0$ or $\mathcal{S}(e, i) = 0$, $\mathcal{S}'(e, i) \neq 0$ for some $i \in \mathcal{I}$. (ii) There exists an link e such that $m_e^{S'} \neq m_e^S$. The procedure for creating a neighboring state $\mathcal{S}'$ of state $\mathcal{S}$ – the heart of SA – is precisely described in Algorithm 1. According to (7a), links with larger π_e^* should be favored. This is achieved by a specific definition of w_e in Step 1. As discussed in Subsection 6.2, this selection turns out to be better than the standard SA (random) neighbor selection.

5.2 SA algorithm

The SA algorithm for problem MaxMinFlow (4) is based on a standard simulated annealing procedure (see for example [6]) and is described in Algorithm 2. The initial solution $\mathcal{S}^0$ is defined as follows:

$$m^{\mathcal{S}^0}(e, i) = \begin{cases} 1, & e = i \\ 0, & e \neq i \end{cases} \quad e \in \mathcal{E}, i = 1, 2, \dots, E. \quad (9)$$

In other words, in $\mathcal{S}^0$ each compatible set contains just one link, with its MCS set to $m = 1$, i.e., the MCS with the smallest rate.

The parameters critical to SA are specified as follows:

- The initial temperature τ_0 . In SA, at a given temperature τ , the probability for accepting a neighbouring solution that is worse by Δ as compared with the current solution, is given by $p = e^{-\frac{\Delta}{\tau}}$. As a result, the initial temperature is set as $\tau_0 = -\frac{\Delta_{avg}}{\ln p_0}$ where p_0 represents the acceptance probability in the first iteration. Δ_{avg} is the average of Δ computed from several trial neighbor transitions.

- The stopping criterion. The algorithm stops when either of the two following conditions is satisfied: (i) there is no improvement in N consecutive steps (one step corresponds to execution of the outer **while** loop); (ii) the solution acceptance ratio becomes smaller than a given probability p_f .
- The number of steps L . The value of parameter L is related to the size of the problem, see Subsection 6.2.
- The cooling scheme. An adaptive way proposed in [25] is adopted to cool down the temperature:

$$\alpha_k = \max\left\{\frac{1}{2}, e^{-\frac{c\tau_k}{\delta_k}}\right\}, \quad \tau_{k+1} = \alpha_k \tau_k$$

where c is a constant between 0 and 1, and δ_k is the standard deviation of the values of objective function that are accepted at temperature τ_k , see Subsection 6.2.

Algorithm 2 Simulated annealing-based heuristic

```

1:  $k := 0$ ,  $\mathcal{S} := \mathcal{S}^0$ ,  $f := \text{Evaluate}(\mathcal{S}^0)$ ,  $\tau_0 = -\frac{\Delta_{avg}}{\ln p_0}$ ,
    $f^{best} := f$ ,  $\mathcal{S}^{best} := \mathcal{S}^0$ 
2: while stopping criterion not satisfied do
3:    $l := 0$ 
4:   while  $l \leq L$  do
5:      $\mathcal{S}' := \text{Neighbor}(\mathcal{S})$ 
6:      $f' := \text{Evaluate}(\mathcal{S}')$ 
7:      $\Delta f := f - f'$ 
8:     if  $\Delta f \geq 0$  then
9:        $\mathcal{S} := \mathcal{S}'$ ,  $f := f'$ 
10:      if  $f' > f^{best}$  then
11:         $f^{best} := f'$ ,  $\mathcal{S}^{best} := \mathcal{S}'$ 
12:      end if
13:    else if  $\text{random}(0,1) \leq e^{-\frac{\Delta f}{\tau_k}}$  then
14:       $\mathcal{S} := \mathcal{S}'$ ,  $f := f'$ 
15:    end if
16:     $l := l + 1$ 
17:  end while
18:   $\alpha_k := \max\left\{\frac{1}{2}, e^{-\frac{c\tau_k}{\delta_k}}\right\}$ 
19:   $\tau_{k+1} := \alpha_k \times \tau_k$ 
20:   $k := k + 1$ 
21: end while
22: return  $f^{best}$ ,  $\mathcal{S}^{best}$ 

```

Our SA algorithm for MaxMinFlow (4) is given in Algorithm 2 where function $\text{Evaluate}(\mathcal{S})$ returns the optimal objective of (8) for $\mathcal{I} = \mathcal{C}^S$, and function $\text{Neighbor}(\mathcal{S})$ returns a neighboring SA solution obtained with Algorithm 1.

Fig. 1: Network examples and optimal MCS assignment

Table 2: Optimal transmission scheduling schemes

Instance	Optimal transmission scheduling
1	$\{(1,4),(2,9)\}, 1.9\%$; $\{(3,8)\}, 7.7\%$; $\{(2,9),(13,11)\}, 7.7\%$; $\{(2,13)\}, 30.7\%$; $\{(13,10)\}, 7.7\%$; $\{(2,5),(13,12)\}, 7.7\%$; $\{(1,4),(2,5)\}, 15.4\%$; $\{(1,6),(2,9)\}, 13.5\%$; $\{(5,7)\}, 7.7\%$
2	$\{(20,6)\}, 5.8\%$; $\{(20,10)\}, 11.5\%$; $\{(1,15)\}, 2.9\%$; $\{(2,18)\}, 2.9\%$; $\{(3,20)\}, 20.0\%$; $\{(12,5)\}, 11.5\%$; $\{(1,7),(4,19)\}, 8.6\%$; $\{(5,9),(2,12)\}, 14.4\%$; $\{(10,4),(7,11),(16,17)\}, 9.7\%$; $\{(5,16)\}, 2.2\%$; $\{(19,8),(7,11)\}, 2.9\%$; $\{(11,14),(5,16)\}, 4.3\%$; $\{(6,13),(16,17)\}, 2.9\%$; $\{(7,11),(16,17)\}, 0.4\%$;

6 Numerical study

In this section we report an extensive numerical study to illustrate optimal solutions for selected network examples and to show the effectiveness of the proposed SA algorithm by comparing it with the branch-and-price method. All the optimization models and algorithms were implemented in Python 2.7 using solver Gurobi 4.5, and run on a computer with the dual core AMD 2.71GHZ CPU and 1.75GB of RAM.

6.1 Experiment setup and solution examples

The tested networks were generated by placing the nodes randomly in a square area of 500 m by 500 m. The transmitting power for each node is set to 100 mW. The propagation attenuation is computed following the free space path loss model with exponent 4 and the noise power set to $\eta = 7.69 \times 10^{-11}$ mW [26]. For any pair of nodes $i, j \in \mathcal{V}$, the power received at node i from node j is calculated as $P_i = P_j \times d_{ij}^{-4}$ and the distance between the two nodes is denoted by $d_{i,j}$. Shortest-hop paths joining the gateways and the mesh routers (one path per router) are used. As we consider only the links in the selected shortest paths, the generated network topologies form the forests (sets of disjoint trees) so the number of links is always smaller than the number of nodes.

Figure 1 depicts two network instances with optimal MCS assignments. The mesh routers are represented by circles while the gateways are represented by squares. Different types of lines correspond to different MCSs assigned to the links as the result of optimization. Table 2 shows the compatible sets (sets of links) and the proportion of time each compatible set is used. The total time is set to $T = 1$ here, and also for all the subsequent experiments.

6.2 Simulated Annealing

For our SA algorithm, the suggested values for parameters are: $p_0 = 0.99$, $p_f = 0.01$, $N = 5$ and $L = \frac{1}{3}E^2$. Then we analyze how the cooling scheme and parameters θ and ϕ affect the efficiency of SA.

To compare adaptive cooling with $c = 0.05$, $c = 0.3$, $c = 0.8$, and exponential cooling with $\alpha = 0.9$, we plot Figure 2(a) which shows the averaged maximized minimum flow over many trials for a randomly selected network. Here we set $\phi = \theta = 0.5$. We observed that in the trials SA always converged. The adaptive scheme can get better objective and needs less time to converge than the exponential cooling scheme. In the adaptive scheme, a bigger c results in a short time to converge but a worse objective. Considering the trade-off between objective and time, we will use the adaptive scheme with $c = 0.1$ in the sequel.

Fig. 2: Parameter analysis for SA

In Step 1 of Algorithm 1 we are using a biased neighbor selection. It is different from the random selection of a standard simulated annealing process as the links with the largest and the smallest weight w_e are selected. In w_e , the entities π_e^* are optimal dual variables corresponding to equations (8c) of the current solution. Table 3 shows the objectives and the running time for five different networks with the two selection methods. The parameter L controls the number of iterations for a given temperature. Clearly, the bigger the value of L , the better the final objective. We set $L = \frac{1}{3}E^2$ for the biased selection and $L = \frac{1}{2}E^2$ for the random selection, which means that the random selection uses more iterations. Still, the biased selection yields a better objective.

For selecting proper values for θ and ϕ , we have run SA over a set of randomly generated networks and draw the Figure 2(b). Each point in the figure is the “Normalized Max-min flow” calculated in the following three steps. First, compute the average objective values over 10 runs of SA for each pair θ, ϕ and for each network, then normalize the results obtained from the first step with respect to the maximum objective for each network, and, finally, average the results normalized in the second step over all networks for each pair θ, ϕ . We can see that the largest objective is obtained for the pair $\phi = 0.6, \theta = 0.6$. We will use this pair of values for the experiments in the next subsection. The parameters θ, ϕ have little impact on the running time and therefore we omit their analysis here.

6.3 Comparison of algorithms

We implemented our own branch-and-bound algorithm for the branch-and-price (B&P) method to exactly solve

Table 3: Comparison of neighbor selection methods

Strategy	Network	N1	N2	N3	N4	N5
Biased	Objective	3.44	1.57	2.15	1.22	0.52
	Running time	10s	72s	220s	2576s	3857s
Random	Objective	3.37	1.56	2.14	1.20	0.51
	Running time	31s	190s	333s	3567s	6586s

MaxMinFlow (4). The branching rule of B&P is to use the “most fractional” (meaning nearest to 0.5) variable. Each B&P node is divided to the left (the branching variable set to 0) and to the right (the branching variable set to 1) child nodes. The B&P tree is traversed in the depth-first manner, with the right child node visited firstly. When solving a linear relaxation RMP($\mathcal{N}$) (problem (5)) for a B&P node, we construct its initial list of compatible sets using formula (9) (selecting the proper m_e instead of $m = 1$ when this is forced by a fixed binary variable $U_e^{m_e} = 1$ in $\mathcal{N}$), and then by adding all the compatible sets of the parent node with consistent MCSs.

The comparisons are made for the three methods: BP (branch-and-price), SA (simulated annealing) and BP-SA. (BP-SA is BP that uses the result of SA for the lower bound.) Table 4 shows the obtained max-min flow (4a) and the corresponding running time. For BP, the first feasible solution and the final optimal solution are given. For each network example, SA was run 10 times. The best (max), worst (min) and average (mean) objectives are reported.

The following observations are made. (1) BP is able to find exact optima for small networks, Net1 and Net2. Also, SA obtains exact optima for some runs, and provides good average results (with a small gap). (2) BP-SA converges much faster than BP to the optimum. For

Table 4: Comparison of algorithms

Network			BP				SA						BP-SA	
name	$ \mathcal{V} $	$ \mathcal{E} $	first		final		max		min		mean		final	
<i>Net1</i>	10	8	0.6	16s	3.48	442s	3.48	14s	3.22	11s	3.40	13s	3.48	172s
<i>Net2</i>	13	10	2.25	25s	4.15	1102s	4.15	41s	3.2	26s	3.99	32s	4.15	463s
<i>Net3</i>	16	13	2.0	148s	4.08	3h*	4.32	78s	3.82	50s	4.2	62s	4.32	2912s
<i>Net4</i>	20	17	0.47	264s	0.50	3h*	1.6	152s	1.48	108s	1.56	145s	1.56	3h*
<i>Net5</i>	25	21	0.54	736s	0.64	3h*	2.22	395s	2.0	285s	2.15	307s	2.15	3h*

(a) Comparison of the gap

(b) Comparison of the running time

Fig. 3: Comparison of BP and SA for large networks

the case *Net3*, BP-SA is able to find exact optima in the time limit while BP cannot. (3) For larger networks *Net4* and *Net5*, the quality of the solutions given by SA is much better than for BP and for BP-SA (both with the 3-hour timeout). (4) For all cases, SA delivers much better solutions in less time than the first feasible solution of BP.

To make further BP vs. SA comparisons, we tested another set of networks with the number of nodes is between 20 and 40. For these networks BP is not capable of finding an optimal solution in 3 hours. The results are shown in Figure 3. In particular, Figure 3(a) shows the curves of GAP_BP and GAP_SA. The value of GAP_BP expresses the ratio $100 \times (\text{final objective} - \text{upper bound}) / (\text{upper bound})$, while GAP_SA represents the ratio $100 \times (\text{mean objective} - \text{upper bound}) / (\text{upper bound})$. The upper bounds are the solutions of the root B&P node, i.e., of the full linear relaxation of MaxMinFlow (4). The value of “final objective” is obtained by BP (run for 3 hours). The value of “mean objective” is the average from 10 SA runs. We observe that SA gives much smaller gap than BP (running for three hours). What is more, SA takes much less time than 3 hours to converge. The running time for SA and for obtaining the first feasible solution of BP is shown in Figure 3(b) for the same networks as for Figure 3(a).

We observe that SA needs much less time than BP, even when compared with the time for obtaining the first feasible solution of BP.

7 Conclusions

In this paper we have studied a flow optimization problem in WMNs by jointly optimizing link rate assignment and transmission scheduling. The major challenge is posed by the assumed requirement that each link uses a fixed (but link-specific and optimized) MSC throughout the entire considered period of network operation. In contrast to WMN optimization problem settings that assume dynamic MSC assignment, generating compatible sets for this case requires a branch-and-price algorithm that is not time-efficient. To construct an effective heuristic algorithm, we utilize a specific problem feature (implied by a property of LP) that E compatible sets are sufficient at an optimum. This characterization is the core in designing our original simulated annealing-based algorithm. The numerical study shows that the heuristic is efficient. Also, by embedding the heuristic solution into B&P, the process of approaching the global optimum can be significantly accelerated. Furthermore, the heuristic can be applied to other optimization problems in WMNs with other objective func-

tions. In fact, the proposed SA algorithm is a heuristic way for column (compatible set in our case) generation in a mixed-integer programming or a linear programming problem. Therefore, it can be applicable whenever the compatible set is used in optimization problems.

Acknowledgements This work was supported by the EC Marie Curie Actions project MESH-WISE (FP7-PEOPLE-2012-IAPP-324515) and Excellence Center at Linköping-Lund in Information Technology (ELLIIT). The research of Michał Pióro and Yuan Li was also supported by National Science Centre (Poland) under grant 2011/01/B/ST7/02967 and by Chinese Scholarship Council, respectively.

References

1. Akyildiz, I. & Wang, X. (2005). A survey in wireless mesh networks. *IEEE Communications Magazine*, 43(9), 23–30.
2. Pathak, P. H. & Dutta, R. (2011). A survey of network design problems and joint design approaches in wireless mesh networks. *IEEE Communications Surveys Tutorials*, 13(3), 396–428.
3. Ashraf, U., Abdellatif, S., & Juanole, G. (2011). Route selection in IEEE 802.11 wireless mesh networks. *Telecommunication Systems*, 1–19.
4. Vergados, D., Sgora, A., Vergados, D., Vouyioukas, D., & Anagnostopoulos, I. (2012). Fair TDMA scheduling in wireless multihop networks. *Telecommunication Systems*, 50, 181–198.
5. Capone, A., Carello, G., Filippini, I., Gualandi, S., & Malucelli, F. (2010). Routing, scheduling and channel assignment in wireless mesh networks: Optimization models and algorithms. *Ad Hoc Networks*, 8(6), 545–563.
6. Pióro, M., Zotkiewicz, M., Staehle, B., Staehle, D., & D.Yuan (2011). On Max-Min Fair Flow Optimization in Wireless Mesh Networks. *Ad Hoc Networks*, doi:10.1016/j.adhoc.2011.05.003.
7. Grace, K., Stine, J., & Durst, R. (2005). An approach for modestly directional communications in mobile ad hoc networks. *Telecommunication Systems*, 28, 281–296.
8. Zotkiewicz, M., & Pioro, M. (2013). Exact approach to reliability of wireless mesh networks with directional antennas. *Telecommunication Systems*, doi:10.1007/s11235-013-9829-4.
9. Desaulniers, G., Desrosiers, J., & Solomon, M. (2005). *Column Generation*. Springer-Verlag.
10. Lasdon, L.S. (1970). *Optimization Theory for Large Systems*. Macmillan London.
11. Montemanni, R. and Leggieri, V. (2010). A branch and price algorithm for the minimum power multicasting problem in wireless sensor networks. *Mathematical Methods of Operations Research*, 74(3), 327–342.
12. Gupta, P. & Kumar, P. (2000). The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), 388–404.
13. Behzad, A. & Rubin, I. (2003). On the performance of graph-based scheduling algorithms for packet radio networks. In *Proceeding of IEEE GLOBECOM*, San Francisco, USA.
14. Huang, X. & Bensaoou, B. (2001). On max-min fairness and scheduling in wireless ad-hoc networks: Analytical framework and implementation. In *Proceeding of ACM MobiHoc*, Long Beach, CA, USA.
15. Leith, D. J., Cao, Q., & Subramanian, V. G. (2012). Max-min fairness in 802.11 mesh networks. *IEEE Transaction on Networking*, 20(3), 756–769.
16. Ahmed, I., Mohammed, A., & Alnuweiri, H. (2013). On the fairness of resource allocation in wireless mesh networks: a survey. *Wireless Networks*, 19(6), 1451–1468.
17. Staehle, B., Staehle, D., & Pries, R. (2009). Effects of link rate assignment on the max-min fair throughput of wireless mesh networks. In *Proceeding of 21st International Teletraffic Congress*, Paris, France.
18. Cai, D. W. H., Tan, C. W., & Low, S. H. (2012). Optimal max-min fairness rate control in wireless networks: Perron-Frobenius characterization and algorithms. In *IEEE INFOCOM*, Orlando, FL, USA.
19. X. Lin, & N. B. Shroff (2004). Joint rate control and scheduling in multihop wireless networks. In *Proceeding of IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas.
20. Avallone, S., Delia, F., & Ventre, G. (2012). A new channel, power and rate assignment algorithm for multi-radio wireless mesh networks. *Telecommunication Systems*, 51, 73–80.
21. Passos, D., & Albuquerque, C. V. N. (2012). A Joint Approach to Routing Metrics and Rate Adaptation in Wireless Mesh Networks. *IEEE/ACM Transaction on Networking*, 20(4), 999–1009.
22. Ancillotti, E., Bruno, R., & Conti, M. (2008). Experimentation and performance evaluation of rate adaptation algorithms in wireless mesh networks. In *Proceedings of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. New York, USA.
23. Xia, D. & Fu, Q. (2011). An experimental study on performance comparison of rate adaptation and fixed rate in IEEE 802.11g. In *Australasian Telecommunication Networks and Applications Conference*, Melbourne, Australia.
24. Pioro, M., & Medhi, D. (2004). Routing, flow, and capacity design in communication and computer networks. Morgan Kaufman.
25. Perrin, G., Descombes, X., Zerubia, J. (2005). Energy minimization methods in computer vision and pattern recognition. Springer Berlin Heidelberg.
26. Goldsmith, A. (2005). Wireless Communication. Cambridge University Press.

A Appendix (proof of Proposition 1)

Consider an arbitrary optimal solution α , ζ_e ($e \in \mathcal{E}$), π_e ($e \in \mathcal{E}$), β_e^m ($e \in \mathcal{E}, m \in \mathcal{M}$) of the dual problem DP($\mathcal{N}$). Define:

- $\hat{\beta}_e = \max \{\beta_e^m : m \in \mathcal{M}\}$, $e \in \mathcal{E}$
- $\hat{\beta}_e^i = \beta_e^m$ if $e \in \mathcal{C}_i$ and $r_{ei}^m = 1$, $e \in \mathcal{E}, i \in \mathcal{I}$ (recall that $r_{ei}^m = 1$ if MCS m is used for link e in compatible set $\mathcal{C}_i$)
- $\hat{\beta}_e^i = 0$ if $e \notin \mathcal{C}_i$ (then $r_{ei}^m = 0$ or all $m \in \mathcal{M}$)
- $\alpha = \max \{\sum_{e \in \mathcal{E}} \pi_e B_{ei} - \sum_{e \in \mathcal{E}} \hat{\beta}_e^i : i \in \mathcal{I}\}$
- i_0 is the index of a compatible set with $\alpha = \sum_{e \in \mathcal{E}} \pi_e B_{ei_0} - \sum_{e \in \mathcal{E}} \hat{\beta}_e^{i_0}$
- $\hat{W}$ is the minimum of the dual function (6a).

With these definitions the following holds

$$\hat{W} = (\alpha + \sum_{e \in \mathcal{E}} \hat{\beta}_e) \cdot T$$

so that

$$\begin{aligned}\frac{\hat{W}}{T} &= \sum_{e \in \mathcal{E}} \pi_e B_{ei_0} - \sum_{e \in \mathcal{E}} \hat{\beta}_e^{i_0} + \sum_{e \in \mathcal{E}} \hat{\beta}_e \\ &= \sum_{e \in \mathcal{E}} \pi_e B_{ei_0} + \sum_{e \in \mathcal{E}} (\hat{\beta}_e - \hat{\beta}_e^{i_0}).\end{aligned}$$

Since, by definition $\hat{\beta}_e \geq \hat{\beta}_e^{i_0}$, $e \in \mathcal{E}$, we can write that

$$\frac{\hat{W}}{T} \geq \sum_{e \in \mathcal{E}} \pi_e B_{ei_0}$$

which shows that the minimum of the dual function, $\hat{W}$, is attained also for the original values of α , π_e ($e \in \mathcal{E}$) and for $\beta_e^m = 0$ ($e \in \mathcal{E}, m \in \mathcal{M}$), and hence for $\zeta_e = 0$ ($e \in \mathcal{E}$). $\square$