Energy-Efficient WLANs with Resource and Re-association Scheduling Optimization

Chuan Xu, *Member, IEEE,* Jiajie Wang, Zuqing Zhu, *Senior Member, IEEE,* and Dusit Niyato, *Fellow, IEEE*

***Abstract*—Recently, a number of WiFi APs have been densely deployed to provide widely-available, high-performance Internet services. As such, an energy efficiency issue becomes crucial toward the design of green wireless local area networks (WLANs). In this paper, we propose a resource and re-association scheduling algorithm (referred to RAS) based on Benders' Decomposition to reduce the energy consumption. In particular, we endeavor to aggregate WLAN users on the small number of APs and turn off many APs without compromising users' quality of experience (QoE) and system coverage. We conduct the analysis by using real trace data and formulate the energy minimization as the mixed integer nonlinear programming (MINLP) problem. We then transform and solve the original problem through the RAS algorithm. For practical implementation, we further propose the fast RAS (Fast-RAS) algorithm to relax the binary integer constraints and transform the MINLP problem into the nonlinear programming (NLP) problem. The relaxed problem then can be solved by using Feasible Pump algorithm with the reduced computational complexity. We evaluate the performance of RAS and Fast-RAS algorithms via extensive simulations. The results demonstrate that the Fast-RAS algorithm can achieve up to 20% improvement of energy saving comparing with existed methods.**

***Index Terms*—Energy efficient WLAN, resource scheduling, user re-association, MINLP optimization.**

I. Introduction

WIFI has been widely deployed in numerous places. Hence, it becomes one of the most popular solutions to provide reliable, low-latency and high-speed Internet connectivity for end users [1]. According to the recent report from Cisco [2], more than 51% mobile traffic is offloaded from cellular networks to both public and residential WiFi access points (APs), and the number will increase to 55% by 2020. Furthermore, to carry over 50% Internet traffic, the number of WiFi APs will grow sevenfold from 64 million in 2015 to 432 million in 2020. Actually, to offer enough capacity that meets the users' demand at peak hours, the density of APs is much higher than that normally needed for coverage only. Additionally, during the off-peak period, as the capacity demand declined sharply in which the utilization of most of APs reduces to a very low level or even zero [3], [4]. However, the power consumption of an idle AP is about 80% of that at full load [5], which leads to serious energy wastage problem.

C. Xu, and J. Wang are with School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China. E-mails: xuchuan@cqupt.edu.cn, wangjjcqupt@gmail.com.

Z. Zhu is with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China. (e-mail: zqzhu@ieee.org).

D. Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: dniyato@ntu.edu.sg).

To achieve green WLANs, the main devices, the APs, which consume 70%-80% energy of WLAN [6], is the focus of improving energy efficiency. This objective can be achieved through AP's transmission power control [7] or switching off/sleeping redundant APs dynamically to avoid low-utilization or idle APs from consuming energy [8]. The switching-off strategy has been widely adopted. For example, the resource-on-demand strategy was proposed for dense WLANs. Likewise, the green clustering algorithm was introduced to initiate a cycle of estimating user demand and performance to power on or off APs [4]. Furthermore, through real traffic analysis, the authors in [9] proposed a simple model to study the AP switching frequency and energy saving. They also presented a detailed investigation of AP turn-off threshold and hysteresis window settings. However, due to user's high mobility, the methods that rely on historical user behavior may fail to accommodate the dynamic network conditions.

Moreover, the real-time monitoring scheme has been introduced to collect information for energy management. The authors in [10] conducted frequent data traffic monitoring on APs. The authors then implemented an automatic sleep control strategy to control the state changes of APs. Similarly, through the centralized control framework, the actual network conditions, i.e., user density and traffic patterns, are monitored and used to tune the energy consumption through a flexible energy-saving decision algorithm [11]. A cooperative energy management method is proposed to schedule wireless resources among gateways based on real-time monitoring [12], and implemented in federated WiFi networks. Nevertheless, the additional energy consumption increase and user's QoE are not considered in the optimization of user re-association in those methods.

Undeniably, switching off low-utilization APs is effective to reduce the energy consumption of green WLANs. However, it may lead to distinct performance degradation for mobile users. Therefore, switching off more APs to save energy, user's QoE must be considered firstly and the effectiveness of an energy saving scheme strongly depends on two factors: 1) which APs are selected to turn off while not sacrificing the user' QoE and system coverage; and 2) the users that associate to the powered off APs need to re-associate to other running APs, as a user can connect to multiple APs, the user re-associates to a suitable AP can minimize the additional energy consumption, which is often ignored in existing methods.

In this paper, aiming to reduce the energy consumption, we seek to aggregate users on fewer APs and to turn off APs. We also optimize the user re-association to minimize total energy consumption while not sacrificing the demands of users' QoE

and system coverage. Unlike existing methods, we take into account the energy increase in the user re-association. With the realistic model of the energy efficiency of APs obtained through real trace data analysis, we formulate the energy efficiency problem as a mixed integer non-linear programming (MINLP) problem, then introduce a resource and re-association scheduling algorithm (RAS) derived from classical Benders' Decomposition to solve the MINLP problem. To reduce the computational complexity of the MINLP problem, we relax the binary integer constraints and transform the master problem into a nonlinear programming (NLP) problem. And then, we propose a fast RAS (Fast-RAS) algorithm based on Feasible Pump to solve the NLP problem within the acceptable computational time.

We validate the performance of the RAS and Fast-RAS algorithms via extensive simulations, and the results demonstrate that the Fast-RAS algorithm can obtain the suboptimal solution close to that of the RAS algorithm. Besides, the Fast-RAS algorithm can reduce the computational time significantly. Moreover, compared with existing methods, the Fast-RAS algorithm achieves up to 20% improvement of energy saving while obtaining much better system coverage and throughput. Especially, when overfull users with high throughput demand generate traffic close to the system's maximum capacity, the Fast-RAS algorithm can still reduce nearly 10% energy consumption, which is much better than the other methods. Note here that the Fast-RAS algorithm can be easily implemented in a software defined networking (SDN) based WLAN systems with a centralized controller to support real-time network monitoring, user re-association management, and AP control. We summarize the contributions of the paper as follows.

- To the best of our knowledge, this paper is the first to model and solve the energy efficiency problem of WLAN systems integrated with the optimization of user re-association to reduce the increase of energy consumption on running APs while meeting the diverse requirements of user's QoE and system coverage.
- Based on real trace data analysis, we model the energy efficiency of an AP device and formulate the energy efficiency problem as a mixed integer non-linear programming (MINLP) problem. Then, derived from Benders' Decomposition, we propose a resource and re-association scheduling algorithm to solve the MINLP problem to obtain the optimal solutions.
- To reduce the computational complexity of the MINLP problem, we relax the binary integer constraints and transform the master problem into an NLP problem. We prove that the solution of the NLP problem is a subset of the optimal solution of the MINLP problem. Then, we propose the Fast-RAS algorithm based on Feasible Pump to solve the NLP problem with reasonable computational complexity.

The rest of the paper is organized as follow. Section II reviews related works. The system model and problem formulation are described in Section III. In Section IV, we present the RAS algorithm developed based on Benders' Decomposition. In Section V, we present the relaxation of the binary integer constraints to simplify the MINLP problem and introduce the Fast-RAS algorithm. The performance of the RAS and Fast-RAS algorithms is evaluated through simulations in Section VI, followed by conclusions in Section VII.

II. Related Works

The idea of resource-on-demand (RoD) strategy for dense WLANs was first proposed by Amit *et al.* [4], and a practical and elegant RoD strategy with green clustering was implemented to form clusters of APs that are close to each other. Specifically, signal attenuation and packet loss rate were selected as the clustering standard to maintain the effective coverage of the network when reducing the redundant idle APs. In [13], the work was extended to account for the case when the APs do not completely overlap their coverage areas, based on single-queue and two-queue analysis, some APs can be switched off to save energy when the maximum capacity is not needed. Furthermore, through investigating users' behavior in real dense WLANs, Debele *et al.* [9] formulated a stochastic characterization and proposed a simple model to design the RoD strategy, which is used to evaluate the AP activity and inactivity periods, AP switching frequency, and energy saving. However, these clustering methods and mechanisms which mainly depend on historical user behavior may fail to accommodate with highly-dynamic network conditions.

Moreover, some researchers try to change the state of APs among finer modes, e.g. busy, idle and closed/sleep states, according to real-time data monitoring on each AP, and design cooperative sleep control algorithms to achieve energy saving [10]–[12], [14]. The authors in [10] performed frequent data traffic monitoring on APs, and an automatic sleep control strategy was implemented to control the state changes of APs. Similarly, through the centralized control framework, the actual network conditions in terms of both user density and traffic patterns are monitored and used to tune the energy consumption through a flexible energy-saving decision algorithm [11]. However, user re-association is not taken into account when switching off idle APs in these methods. Thus, the users' QoE can not be guaranteed. Then, the authors in [12] proposed a centralized control framework, which allows the cooperation through the monitoring of local wireless resource and the triggering of offloading requests toward other APs in federated WiFi networks to realize the policy of AP switching and user offloading. In [14], an energy balancing AP coordination mechanism called Crowd-AP was proposed. The mechanism controls each AP to gather information of wireless network and share the information with its neighbor APs. This mechanism can also perform user's re-association, but without the consideration of AP's capacity. Moreover, the bandwidth and storage of APs will be exhausted by a large amount of additional data exchange.

Recently, the idea of centralized control has been introduced to achieve high energy efficiency. With the development of the SDN technology [15], [16], real-time network monitoring and centralized control can be achieved to solve the energy efficiency problem in WLANs. Amokrane *et al.* [17] proposed a flow-based management framework to improve

energy efficiency in campus networks. To support users' demands and mobility, an online flow-based routing approach was introduced to reconfigure the existing flows dynamically with link rate adaptation. Chen *et al.* [18] proposed a QoS-aware AP energy saving mechanism using SDN to reduce energy consumption without affecting user's QoS requirements. Meanwhile, the optimization of resource allocation and user association have become the new approaches to improve network performance. For example, Nessrine *et al.* [19] proposed a solution of user association and inter-cell interference coordination to maximize the network utility of LTE cellular networks. Similarly, Fan *et al.* [20] optimized the user association in WLAN/cellular integrated network to achieve a tradeoff between the throughput and power of users' equipments.

Therefore, to improve the energy efficiency of WLANs, in this paper, we attempt to optimize the resource and user re-association with the consideration of energy consumption incurred when the user is re-associated to a new AP. We also employ SDN-based architecture to implement our algorithm. We formulate the energy efficiency problem as an MINLP problem, and introduce Benders' Decomposition to solve it. Benders' Decomposition is proposed by Benders [21] and extended by Geoffrion [22]. The Benders' Decomposition has a great advantage in solving complex mathematical problems such as mixed integer programming and stochastic programming. Kheirkhah *et al.* [23] proposed an improved Benders' Decomposition algorithm for a capacitated vehicle routing problem, which is divided into the network interdiction master problem and the vehicle routing selection subproblem. Qian *et al.* [24] studied a joint optimization of BS association and power control in heterogeneous cellular networks based on Benders' Decomposition. The objective is to achieve that good communication quality is retained on each link at the minimum cost. Nasri *et al.* [25] proposed an efficient solution approach based on Benders' Decomposition to solve a network-constrained unit commitment problem under uncertainty. However, none of them tries to optimize the energy consumption for the deployment of APs while guaranteeing the communication quality for the users. Our paper focuses on utilizing Benders' Decomposition to obtain the globally optimal solution of the energy efficiency problem.

III. Resource and Re-association Scheduling for Energy-Efficient WLANs

A. Energy efficiency management in SDN based WLANs

The problem of maximizing energy efficiency in a typical WLAN scenario with dense APs is illustrated in Fig. 1. During the off-peak period, there is only small number of users connected to AP_3 and AP_4, and the utilization of these two APs is low. Therefore, the users on AP_3 and AP_4 can be re-associated to other APs without affecting user's QoS given that the other APs have sufficient bandwidth to support these switching users, and thus AP_3 and AP_4 can be turned off to save the energy. Moreover, through further optimizing the re-association of users to suitable APs, the increment of energy consumption on the APs can be minimized.

By using the SDN and NFV technologies in heterogeneous wireless networks, the resource scheduling optimization of

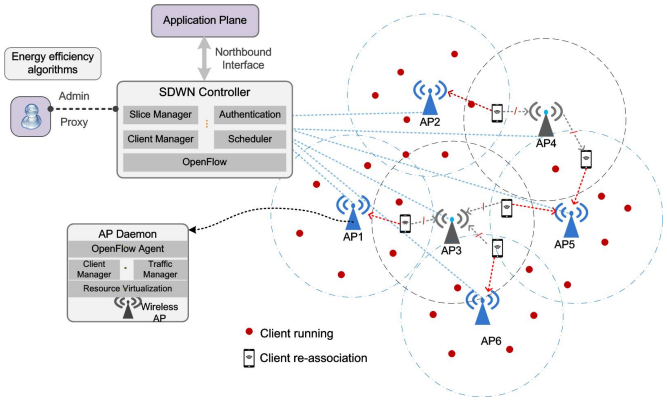


Fig. 1: The energy efficiency in dense SDN based WLANs.

energy management in WLANs becomes feasible [26], [27]. Therefore, we propose a novel Software-defined Wireless Network (SDWN) architecture [16]. Firstly, the centralized architecture of SDN provides a global view of the network, and the control plane and data plane decoupling supports real-time monitoring as well as flexible control of the APs, which enables a centralized network-side AP association mechanism. Moreover, the abstraction of network functionality achieved from the underlying physical infrastructure can support seamless handover among APs using different channels for mobile users and avoid the re-association overhead [28], [29]. As shown in Fig. 1, the network entities are described as follows.

1) SDWN Controller: The SDN controller enables network applications to orchestrate the underlying physical wireless network entities and services. The SDN controller provides a set of interfaces (the northbound interface) to the applications and translates their requests into a set of commands (the southbound interface) to the network devices. It maintains a view of the network components including users, APs, applications, and OpenFlow switches, and performs the centralized functions including user authentication, creating, migrating and releasing of a virtual access point (VAP) for per-user, network performance monitoring and resources scheduling. Here, the VAP can be formed over different physical APs.

2) AP Daemon: An AP Daemon runs on the physical APs and executes the command from the SDN controller to orchestrate the wireless networks, measure, and report the VAP performance of users on APs. Firstly, it works on the WiFi devices to virtualize high-level wireless functions of the IEEE 802.11 MAC protocol for different network slices. Secondly, it maintains VAP and wireless flow transmission rules for each user, dispatches and forwards the traffic flow according to flow-tables. Thirdly, as the status of users changes frequently, it sniffs the wireless data frames for monitoring the performance of VAPs to support a publish-subscribe information system when a particular frame event is triggered.

3) Admin Proxy: The admin proxy provides a set of open programmable interfaces for system administrators. The administrators can create multiple independent network slices, and deploy their routing or scheduling algorithms into network slices. The admin proxy allows the administrator to exert the network control, e.g., load balancing, energy efficiency,

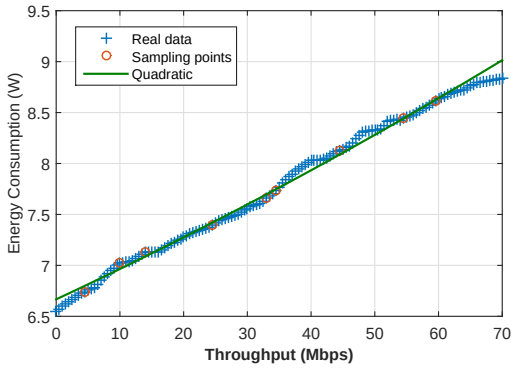


Fig. 2: The fitting curve of AP's energy efficiency.

troubleshooting, and to supports test run evaluation from implementing the customized flow assignment or migration algorithm into the SDWN core module.

B. System Model

1) Network model: As shown in Fig. 1, we consider a centralized WLAN system that every AP is under the central controller. The controller acquires and schedules the network resources to serve mobile users.

- *AP control*: The SDN controller controls the on-off switching and resources of APs. When some APs are unnecessary to be turned on, the controller will turn off them to reduce the energy consumption and improve energy efficiency. When the bandwidth required by the users that are associated with an AP exceeds the total bandwidth available at the AP, the controller will turn on a new AP. The new AP will also support some of the users in its corresponding coverage. We denote an active mode indicator of AP_i by $\alpha_i \in \{0, 1\}$, where $\alpha_i = 1$ if AP_i is on, and $\alpha_i = 0$ otherwise.
- *User association control*: Based on the NFV technology, users in the WLAN system can migrate among APs seamlessly and flexibly. When a set of active APs is selected to be turned off, the users associated with those APs will be re-associated to other APs. Therefore, the controller needs to decide the re-association between all the mobile users and the APs. We denote an association between user j and AP_i by β_{ij}, where $\beta_{ij} = 1$ if user j is associated with AP_i, and $\beta_{ij} = 0$ otherwise. Moreover, since the re-association of the user is constrained by whether the user is in the coverage of the new AP or not, we denote a relation between user j and the coverage of AP_i by $C_{ij} \in \{0, 1\}$, where $C_{ij} = 1$ if user j is within the coverage of AP_i, and $C_{ij} = 0$ otherwise.
- *Transmission*: Since our goal is to reduce the energy consumption of the APs, which are incurred mostly from the wireless signal transmission, we focus on the downlink communication, i.e., from APs to users. We denote the minimum requirement of the active data rate for user j by d_j and assume that d_j is a constant during the resource scheduling optimization period.

2) Energy Consumption of AP: We conduct an experiment in real networks to test the energy efficiency of WiFi devices. We deploy 10 Netgear WNDR 3800 and 4300 devices in a typical application scenario: working at 2.4 GHz with 802.11n mode, 20 MHz HT mode, random channel, and 30 dBm transmission power. We use power meter (TECMAN-TM6) to measure AP's energy consumption.

From the real trace data, we find that when the throughput is higher than 70Mbps, the loss rate suddenly increases from 0.94% shapely, which leads to the decrease of user's QoE. Consequently, we let l_{max} denote the threshold and set it to 70Mbps throughput. If AP's load exceeds the threshold, a new AP will be turned on to support the increasing users' demands.

As shown in Fig. 2, to analyze the energy efficiency of WiFi devices, we randomly sample the experimental data and adopt the Minimum Mean Square Error (MMSE) method to compute and obtain the energy efficiency fitting function. The function can present the trend of energy consumption with the change of throughput on the APs. To improve the fitting accuracy, we increase the sampling rate to 160 samples per second and model the energy consumption function as follows:

$$P(l) = al^2 + bl + c, \tag{1}$$

where l and $P(l)$ denote the AP's throughput and energy consumption with the throughput l, respectively. The values of variables a, b and c are $6.2185e-5$, 0.029173 and 6.667, respectively. The AP's energy consumed is composed of a fixed component and a variable component. The fixed component, denoted by $\sigma_0 = P(0)$, is for the AC/DC conversion, basic circuitry powering, dispersion, etc. The variable component associated with the throughput is given by $f(l)$ as follows:

$$f(l) = al^2 + bl. \tag{2}$$

C. Problem Formulation

Formally, the energy efficiency problem of the WLAN formulated based on a bipartite structure is to allocate a set of mobile users to the minimum number of active APs in which each user must be associated with only one AP. The objective is to minimize the energy consumption of the active APs, while guaranteeing the users' demands without exceeding the maximum capacity of the connections between the users and the APs.

We assume that the WLAN system includes n APs and m users, and denote their sets by $N = \{1, \ldots, n\}$ and $M = \{1, \ldots, m\}$, where AP $i \in N$ and user $j \in M$, respectively. The users' throughput demands matrix is denoted by $d = [d_1, \ldots, d_m]^\top \in R^{m\times 1}$, and the energy consumption of AP_j denoted as p_i, i.e.,

$$p_i = P\left(\sum_{j=1}^{m} \beta_{ij} d_j\right). \tag{3}$$

Based on real-time monitoring, we can initialize the state of users and APs as follows. Let $\alpha^{(0)}$ denote the working status of APs, where $\alpha^{(0)} = [1, \ldots, 1]^\top \in R^{n\times 1}$, $\beta^{(0)}$ denotes the association matrix between users and APs, $\beta^{(0)} = \left[\beta_{ij}\right] \in R^{n\times m}$, $C^{(0)}$ denotes the signal coverage matrix between users and

APs, $C^{(0)} = \left[C_{ij}\right] \in R^{n\times m}$, and $L^{(0)}$ denotes the throughput of all APs, and $L^{(0)} = [l_1, \ldots, l_n]^\top = \beta^{(0)} \cdot d \in R^{n\times 1}$. Moreover, the parameters are given by

$$\begin{aligned} L^{max}&=[l_{max}] \in R^{n\times 1}, \\ \sigma&=[\sigma_0] \in R^{n\times 1}. \end{aligned} \tag{4}$$

Given the definitions of these variables in system model, the resource and re-association scheduling optimization problem can be expressed as follows:

$$\mathfrak{I}_1 : \min_{\alpha_i,\beta_{ij}} \sum_{i=1}^{n} \left\{ P\left(\sum_{j=1}^{m} \beta_{ij} d_j\right)\alpha_i \right\}, \tag{5}$$

$$\text{s.t. } \alpha_i, \beta_{ij} \in \{0,1\}, \quad \forall i \in N, \forall j \in M, \tag{6}$$

$$\sum_{i=1}^{n} \beta_{ij} = 1, \quad \forall j \in M, \tag{7}$$

$$\sum_{i=1}^{n} \alpha_i \beta_{ij} = 1, \quad \forall j \in M, \tag{8}$$

$$0 \le \beta_{ij} \le C_{ij}, \quad \forall i \in N, \forall j \in M, \tag{9}$$

$$0 \le \sum_{j=1}^{m} \beta_{ij} d_j \le l_{max}, \quad \forall i \in N. \tag{10}$$

The objective in (5) indicates the energy consumption of all active APs to be minimized, where $P(l)$ is convex in l. Solving the problem $\mathfrak{I}_1$ means that the corresponding algorithm should return the optimal active AP vector α^* and the user-AP association matrix β^*. The constraint in (6) represents the feasible set of α_i and β_{ij}. The constraints in (7) and (8) ensure that user j can be associated to one AP, and the constraint in (9) ensures that only the user that is within the coverage of the AP can be associated to the AP. The constraint in (10) ensures that the total throughput of every AP is within its transmission capacity.

Since the energy consumption of an AP can be divided into two components, the mixed integer non-linear programming (MINLP) problem $\mathfrak{I}_1$ can be rewritten as $\mathfrak{I}_2$, i.e.,

$$\begin{aligned} \mathfrak{I}_2 : &\min_{\alpha_i,\beta_{ij}} \Phi(\alpha,\beta) \\ &\text{s.t. } \alpha_i, \beta_{ij} \in \{0,1\}, \quad \forall i \in N, \forall j \in M, \\ &\sum_{i=1}^{n} \beta_{ij} = 1, \quad \forall j \in M, \\ &0 \le \beta_{ij} \le C_{ij}, \quad \forall i \in N, \forall j \in M, \\ &0 \le \sum_{j=1}^{m} \beta_{ij} d_j \le l_{max}, \quad \forall i \in N. \end{aligned} \tag{11}$$

where

$$\begin{aligned} &\Phi(\alpha,\beta) \\ &= \sum_{i=1}^{n} f\left(\sum_{j=1}^{m} \beta_{ij} d_j\right) + \sigma_0 \sum_{i=1}^{n} \alpha_i \\ &= f(\beta) + \min_{\alpha} \left\{ \sigma^\top \alpha | \alpha_i \ge \beta_{ij}, \forall j \in M \right\}, \\ &f(\beta) = a \sum_{i=1}^{n} \left(\sum_{j=1}^{m} \beta_{ij} d_j\right)^2 + b \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_{ij} d_j. \end{aligned} \tag{12}$$

Lemma 1 (Convexity): The objective function $\mathfrak{I}_1$ is convex, and the feasible set bounded by the constraint functions (6)-(10) is convex.

Proof: The second term $\sigma_0 \sum_{i=1}^{n} \alpha_i$ in the objective function of $\mathfrak{I}_2$ is linear. We only need to demonstrate the convexity of the first part $f(\beta)$ in the objective function of $\mathfrak{I}_2$. Through introducing the parameter $\lambda \in (0,1)$, we have

$$\begin{aligned} \lambda f\left(\beta^1\right) =& \sum_{i=1}^{n} \left[\lambda a \left(\sum_{j=1}^{m} \beta_{ij}^1 d_j\right)^2 + \lambda b \left(\sum_{j=1}^{m} \beta_{ij}^1 d_j\right)\right], \\ (1-\lambda) f\left(\beta^2\right) =& \sum_{i=1}^{n} \left[(1-\lambda) a \left(\sum_{j=1}^{m} \beta_{ij}^2 d_j\right)^2 \right. \\ &\left. + (1-\lambda) b \left(\sum_{j=1}^{m} \beta_{ij}^2 d_j\right)\right]. \end{aligned} \tag{13}$$

Also, we have

$$\begin{aligned} f\left(\lambda\beta^1 + (1-\lambda)\beta^2\right) = \sum_{i=1}^{m} \left\{ a \left[\sum_{j=1}^{m} \left(\left(\lambda\beta_{ij}^1\right)^2 + \left((1-\lambda)\beta_{ij}^2\right)^2 \right.\right.\right. \\ \left.\left.\left. + 2\lambda(1-\lambda)\beta_{ij}^1\beta_{ij}^2\right) d_j^2 \right] + b \left[\sum_{j=1}^{m} \left(\lambda\beta_{ij}^1 + (1-\lambda)\beta_{ij}^2\right) d_j \right] \right\} \end{aligned} \tag{14}$$

According to the inequality of arithmetic and geometric means, we have $2\beta_{ij}^1\beta_{ij}^2 \le \left(\beta_{ij}^1\right)^2 + \left(\beta_{ij}^2\right)^2$, and the expression in (13) becomes

$$\begin{aligned} &\sum_{i=1}^{m} \left\{ a \left[\sum_{j=1}^{m} \left(\left(\lambda\beta_{ij}^1\right)^2 + \left((1-\lambda)\beta_{ij}^2\right)^2 \right.\right.\right. \\ &\left.\left.\left. +2\lambda(1-\lambda)\beta_{ij}^1\beta_{ij}^2\right) d_j^2 \right] \right. \\ &\left. +b \left[\sum_{j=1}^{m} \left(\lambda\beta_{ij}^1 + (1-\lambda)\beta_{ij}^2\right) d_j \right] \right\} \\ \le& \sum_{i=1}^{m} \left\{ a \sum_{j=1}^{m} \left[\lambda\left(\beta_{ij}^1 d_j\right)^2 + (1-\lambda)\left(\beta_{ij}^2 d_j\right)^2 \right] \right. \\ &\left. +b \left[\sum_{j=1}^{m} \left(\lambda\beta_{ij}^1 + (1-\lambda)\beta_{ij}^2\right) d_j \right] \right\} \\ =& \lambda f\left(\beta^1\right) + (1-\lambda) f\left(\beta^2\right). \end{aligned} \tag{15}$$

From (15), we can obtain that $\left(\lambda\beta^1 + (1-\lambda)\beta^2\right) \le \lambda f\left(\beta^1\right) + (1-\lambda) f\left(\beta^2\right)$. Therefore, the first part of $\mathfrak{I}_2$ is convex and $\mathfrak{I}_1$ is hence convex.

According to the definition of convex function, the linear constraints in (6), (7), (9) and (10) are both convex and concave. As such, the feasible sets bounded by the constraints in (6), (7), (9) and (10) are the convex sets. For the constrains in (8), when one of α and β is determined, the other term constitutes a linear function, and its feasible solutions constitute a convex set. Additionally, the feasible set is convex because of the intersection of the feasible sets bounded by the constraints in (6), (7), (9) as well as (10) and the feasible set bounded by

the constraint in (8). Therefore, the problem $\mathfrak{I}_1$ is the convex optimization, and a globally optimal solution exists. ■

As we can see from (12), the problem $\mathfrak{I}_2$ divides the two decision variables which are α and β in the problem $\mathfrak{I}_1$ into two parts. Specifically, the problem $\mathfrak{I}_2$ is a combination of integer non-linear problem and integer linear problem that represents the original problem. However, it is easier to solve than $\mathfrak{I}_1$, which is a mixed integer non-linear programming problem. However, the problem $\mathfrak{I}_2$ is still complex to solve with complicated AP switching and re-association strategies. Therefore, to solve the NP-hard problem $\mathfrak{I}_2$, we propose an efficient algorithm based on the Bender's Decomposition.

IV. The RAS Algorithm based on Benders Decomposition

A. Benders' Decomposition for the MINLP Problem

Benders' decomposition is an efficient technique in solving certain classes of difficult optimization such as mixed-integer nonlinear programming problems [30]. To avoid handling all variables and constraints of a problem simultaneously, the Benders' Decomposition divides the problem into two subproblems, i.e., the 0/1 integer programming subproblem and the linear programming subproblem with continuous variables.

Note that the problem $\mathfrak{I}_2$ has the same form as the problem in Benders' Decomposition [31]. Therefore, we propose the RAS algorithm based on the Benders' Decomposition to solve the problem $\mathfrak{I}_2$. In particular, the problem $\mathfrak{I}_2$ can be decomposed into the maximization subproblem as expressed in (17) and the master problem as expressed in (18) at the kth iteration:

The primal subproblem is

$$\begin{aligned} \min_{\alpha} \ & \sigma^{\top}\alpha \\ \text{s.t. } & \alpha_i \in \{0,1\}, \forall i \in N, \\ & \alpha_i \geq \beta_{ij}^{(k)}, \forall i \in N, \forall j \in M. \end{aligned} \tag{16}$$

The dual of subproblem is:

$$\begin{aligned} \max_{w} \ & \Psi\left(\beta^{(k)}, w\right) \\ \text{s.t. } & \alpha_i w_z \leq \sigma_0, \forall i \in N, \forall j \in M, \\ & w_z \geq 0, \\ & z = (i-1)\, m + j, \forall i \in N, \forall j \in M, \end{aligned} \tag{17}$$

where $\Psi\left(\beta^{(k)}, w\right) = \sum_{i}^{n}\sum_{j}^{m} \beta_{ij}^{(k)} w_z$, $w \in R^{m\times n\cdot 1}$. According to the duality theorem, we infer that the length of vector w should be equal to the number of elements in the matrix β. We define that $z = (i-1)\, m + j$ (i.e., if m is 10, the w_{11} wil correspond with $\beta_{2,1}$) in the calculation.

The master problem is:

$$\begin{aligned} \min_{\beta,t} \ & f(\beta) + t \\ \text{s.t. } & \beta_{ij} \in \{0,1\}, \forall i \in N, \forall j \in M, \\ & \sum_{i=1}^{n} \beta_{ij} = 1, \quad \forall j \in M, \\ & 0 \leq \beta_{ij} \leq C_{ij}, \quad \forall i \in N, \forall j \in M, \\ & 0 \leq \sum_{j=1}^{m} \beta_{ij} d_j \leq l_{max}, \quad \forall i \in N. \\ & t \geq 0, \\ & \Psi\left(\beta, w^{(p)}\right) \leq t, \forall p = 1, \ldots, k_1, \\ & \Psi\left(\beta, w^{(q)}\right) \leq 0, \forall q = 1, \ldots, k_2, \end{aligned} \tag{18}$$

where $\beta^{(k)}$ is the optimal solution of (18) at the kth iteration and $f(\beta) = a\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\beta_{ij}d_j\right)^2 + b\sum_{i=1}^{n}\sum_{j=1}^{m}\beta_{ij}d_j$. If the subproblem in (17) is bounded for $\beta^{(k)}$, we compute the optimal solution $w^{(k_1)}$ of (17) and add an optimality cut $\Psi\left(\beta, w^{(k_1)}\right) \leq t$ to the master problem (18). Otherwise, obtain an point $w^{(k_2)}$ on the extreme ray and a feasible cut $\Psi\left(\beta, w^{(k_2)}\right) \leq 0$ is added to (18). Obviously, k will be always equal to the sum of k_1 and k_2. At the kth iteration, the master problem (18) will be solved for $\beta^{(k)}$ with all $(k-1)$ constraints which are added in the previous iterations.

At each iteration k, a lower bound $LB^{(k)} = L^{(k)}$ and an upper bound $UB^{(k)} = \min\left\{f\left(\beta^{(k)}\right) + U^{(k)}\right\}$ for the optimal objective value of the problem $\mathfrak{I}_2$ can be calculated based on solutions of (18) and (17), where $L^{(k)}$ and $U^{(k)}$ are the objective values of (18) and (17) respectively.

Lemma 2 (Bound): The upper and lower bounds are tightened at each iteration until they converge to the optimal solution.

Proof: The lower bound $LB^{(k)}$: We prove that $LB^{(k)} = L^{(k)}$ is the lower bound of the optimal objective function $\mathfrak{I}_2$. As shown in (17), $\min\{\sigma^{\top}\alpha\}$ is an LP problem with strong duality. By the strong duality of $\min\{\sigma^{\top}\alpha\}$, the problem $\mathfrak{I}_2$ is equivalent to

$$\begin{aligned} \min_{\beta,w} \ & f(\beta) + \Psi(\beta, w) \\ \text{s.t. } & \beta_{ij} \in \{0,1\}, \quad \forall i \in N, \forall j \in M, \\ & \sum_{i=1}^{n} \beta_{ij} = 1, \quad \forall j \in M, \\ & 0 \leq \beta_{ij} \leq C_{ij}, \quad \forall i \in N, \forall j \in M, \\ & 0 \leq \sum_{j=1}^{m} \beta_{ij} d_j \leq l_{max}, \quad \forall i \in N. \end{aligned} \tag{19}$$

Let (α^*, β^*) be the optimal solution of $\mathfrak{I}_2$. Since $\Psi(\beta, w)$ is the objective function for the dual formulation of the primal problem (17), it follows that $f(\beta^*) + \sigma^{\top}\alpha^* \geq L^{(k)} = f\left(\beta^{(k)}\right) + t^{(k)}$, where $\left(\beta^{(k)}, t^{(k)}\right)$ is the solution of (18) at the kth iteration. Therefore, the objective value of (17) as $L^{(k)}$ is the lower bound of function $\mathfrak{I}_2$.

The upper bound $UB^{(k)}$: We prove that $UB^{(k)} = f\left(\beta^{(k)}\right) + U^{(k)}$ is the upper bound of function $\mathfrak{I}_2$. Firstly, $U^{(k)}$ is either finite or infinite depending on the boundary of (17). If $U^{(k)} = +\infty$, then $UB^{(k)} = f\left(\beta^{(k)}\right) + U^{(k)} = +\infty$. In this case, it is meaningless to tighten the upper bound of $\mathfrak{I}_2$. Therefore, we focus on the case when (17) is bounded and $U^{(k)} < +\infty$. Because of the strong duality of (17), we can obtain the relationship as follows:

$$L^{(k)} = \Psi\left(\beta^{(k)}, w^{(k)}\right) \leq \sigma^{\top}\alpha^* \leq \sigma^{\top}\alpha^{(k)} = U^{(k)}, \tag{20}$$

where, $\alpha^{(k)}$ is the optimal solution to $\min\{\sigma^\top \alpha\}$ with $\beta^{(k)}$. It can be obtained from

$$\begin{aligned} UB^{(k)} &= f\left(\beta^{(k)}\right) + U^{(k)} \\ &= f\left(\beta^{(k)}\right) + \sigma^\top \alpha^{(k)} \geq f\left(\beta^*\right) + \sigma^\top \alpha^*. \end{aligned} \tag{21}$$

The proof is done. ■

B. The RAS Algorithm

The details of the RAS algorithm designed based on the Benders' Decomposition is given below. Based on real-time monitoring of the network status, we initialize the active AP vector $\alpha^{(0)}$ and the association matrix $\beta^{(0)}$, and we set t to 0. First, we solve the subproblem in (17) with $\beta^{(0)}$ to obtain the optimal solution $w^{(0)}$. Then, with a given $w^{(0)}$, we can construct a new function $t^*(\beta) = \Psi\left(\beta, w^{(0)}\right)$ and compare t to the upper and lower bounds of $t^*(\beta)$. If it is bounded, $w^{(0)}$ is an extreme point in the feasible set of (17), and adds an optimality cut $\Psi\left(\beta, w^{(0)}\right) \leq t$ to the master problem as the new constraint. Otherwise, $w^{(0)}$ is a point on the extreme ray and adds a feasible cut $\Psi\left(\beta, w^{(0)}\right) \leq 0$ to (18). With the new constraint, the RAS algorithm computes the optimal solution of the problem in (18) to obtain $\beta^{(1)}$ and the value of t as $t^\dagger$. Then, the algorithm uses $\beta^{(1)}$ to solve the subproblem in (17) and construct the new constraint by the boundness of $t^*(\beta)$ with $w^{(1)}$ again. The RAS algorithm repeats this procedure until an optimal solution of $\mathfrak{I}_2$ is found. If $UB^{(k)} - LB^{(k)} \leq \tau$ at the kth iteration, the RAS algorithm will terminate. To avoid falling into a large number of redundant iterations, the parameter τ is generally set to a sufficiently small value, e.g., 10^{-4}. Finally, we obtain the optimal association matrix β^* and the AP state matrix α^*, which yield the minimum energy consumption of the WLANs.

Aiming to find optimal energy-saving associations between users and APs for minimizing the number of active APs, we consider the AP switching problem and user re-association problem jointly to achieve optimal energy saving.

THEOREM *1 (Convergence):* Within the finite number of iterations, the RAS algorithm converges to a globally optimal solution to the problem $\mathfrak{I}_2$.

Proof: For the second part of $\mathfrak{I}_2$: $\min\left\{\sigma^\top \alpha | \alpha_i \geq \beta_{ij}, \forall j \in M\right\}$, we can obtain the maximum number of no-load APs, i.e., the APs without traffic load, through enumerating all association patterns. However, in general, the solution is not unique, as different association patterns may lead to the same number of no-load APs. Let (α^*, β^*) be the optimal solution of the problem $\mathfrak{I}_2$ and $\left(\tilde{\alpha}, \tilde{\beta}\right)$ be the solution of the problem $\mathfrak{I}_2$'s second part, we can reach

$$\begin{aligned} \alpha^* &= \tilde{\alpha}, \\ \Phi\left(\alpha^*, \beta^*\right) &\geq \Phi\left(\tilde{\alpha}, \tilde{\beta}\right). \end{aligned} \tag{22}$$

For the first part of the problem $\mathfrak{I}_2$: $\min f(\beta)$, we can find an association strategy based on the solution of the second part which refers to the minimum energy consumption of the WLANs. Then, we can confirm the following conditions:

$$\begin{aligned} \beta^* &= \tilde{\beta}, \\ \Phi\left(\alpha^*, \beta^*\right) &\leq \Phi\left(\tilde{\alpha}, \tilde{\beta}\right). \end{aligned} \tag{23}$$

Algorithm 1 The RAS algorithm

Initialization: $k = 0, UB^{(0)} = +\infty, LB^{(0)} = 0, t = 0$.

```
while UB^(k) − LB^(k) > τ do
    if k=0 then
        Set β^(0) and C^(0) according to the network status.
    else
        Solve the master problem in (18) to obtain the optimal solution β^(k), t† and the lower bound LB^(k).
    end if
    Solve the subproblem in (17) with β^(k) to obtain the upper bound UB^(k) as min{f(β^(k) + U^(k))} and the optimal solution w^(k).
    Define function t*(β) = Ψ(β, w^(k)) and obtain the upper bound of t as t*_max and the lower bound of t as t*_min.
    if t*_min ≤ t† ≤ t*_max then
        It is bounded, and we add the constraint Ψ(β, w^(k1)) ≤ t to the master problem in (18).
    else
        It is unbounded, and we add the constraint Ψ(β, w^(k2)) ≤ 0 to the master problem (18).
    end if
    k ← k + 1.
end while
Select the optimal selected APs and switch them to sleep mode according to AP's active matrix α through solving the primal subproblem in (16) with β^(k). Return (α, β^(k)) as the optimal solution of the problem 𝔍2.
```

Therefore, it is effective and beneficial to divide the problem $\mathfrak{I}_2$ into two parts based on the principle of Benders' Decomposition to obtain the optimal solution α^* and β^* satisfying

$$\begin{aligned} \alpha^* &= \tilde{\alpha}, \\ \beta^* &= \tilde{\beta}, \\ \Phi\left(\alpha^*, \beta^*\right) &= \Phi\left(\tilde{\alpha}, \tilde{\beta}\right), \end{aligned} \tag{24}$$

to minimize the energy consumption. ■

V. Fast-RAS Algorithm

As a mixed-integer nonlinear programming problem, the master problem in (18) of the RAS algorithm dominates the computational complexity at each iteration. To address this issue, we propose an improved algorithm, namely, the Fast-RAS algorithm, to simplify the master problem (18). Through relaxing the binary integer constraints, the Fast-RAS problem transforms the master problem in (18) into a nonlinear programming problem and then employs the Feasible Pump algorithm to solve the nonlinear problem to obtain the optimal solution efficiently. First, we rewrite the master problem in (18) as follows:

$$\begin{aligned} \min_{\beta, t} \quad & f(\beta) + t \\ \text{s.t. } & 0 \leq \beta_{ij} \leq 1, \forall i \in N, \forall j \in M, \\ & \sum_{i=1}^{n} \beta_{ij} = 1, \quad \forall j \in M, \end{aligned}$$

$$
\begin{aligned}
&0 \le \beta_{ij} \le C_{ij}, \quad \forall i \in N, \forall j \in M,\\
&0 \le \sum_{j=1}^{m} \beta_{ij} d_j \le l_{max}, \quad \forall i \in N,\\
&t \ge 0,\\
&\Psi\left(\beta, w^{(p)}\right) \le t, \forall p = 1, \ldots, k_1,\\
&\Psi\left(\beta, w^{(q)}\right) \le 0, \forall q = 1, \ldots, k_2,
\end{aligned} \tag{25}
$$

where the discrete domain $\{0, 1\}$ for the variable β_{ij} is replaced by a continuous variable, the interval of which is $[0, 1]$.

THEOREM *2:* We assume that $\left(\hat{\beta}, \hat{t}\right)$ is an arbitrary feasible solution of the problem in (25). The optimality cut or feasibility cut is generated through solving the subproblem in (17) with $\hat{\beta}$ within the optimal solution of the problem $\Im_2$, (α^*, β^*), from the remaining feasible set of (25).

Proof: According to the RAS algorithm, we can derive that if the subproblem in (17) is bounded with $\hat{\beta}$, an optimal solution $\hat{w}$ of (17) can be obtained and an optimality cut

$$\Psi\left(\beta, \hat{w}\right) \le t, \tag{26}$$

is added to the integer-relaxed master problem (25). Otherwise, if the subproblem in (17) is unbounded with $\hat{\beta}$, a point $\hat{w}'$ on the extreme ray of (17) is found and a feasible cut

$$\Psi\left(\beta, \hat{w}'\right) \le 0, \tag{27}$$

is added to the integer-relaxed master problem in (25).

Two different conditions lead to the different new constrains. In the following proof, we show that the optimal solution (α^*, β^*) of the problem $\Im_2$ does not violate the constraint neither in (26) nor in (27). Considering w^* to be the optimal solution of the subproblem in (17) with β^*. The corresponding optimal objective function value, denoted by t^*, is given as $t^* = \Psi\left(\beta^*, w^*\right)$.

On the one hand, when the subproblem in (17) is bounded with $\hat{\beta}$, we suppose that (β^*, t^*) violates (26), i.e., $\Psi\left(\beta^*, \hat{w}\right) > t^*$. This contradicts the fact that w^* is the optimal solution to (17) with β^*, and thus t^* should be the maximum value of (17) with β^*. Hence, the optimality cut (26) cannot be violated by β^*.

On the other hand, when the subproblem in (17) is unbounded with $\hat{\beta}$, we suppose that β^* violates the feasibility cut (27), i.e., $\Psi\left(\beta^*, \hat{w}'\right) > 0$. Since $\hat{w}'$ is a point on the extreme ray of the feasible set of the problem in (17), $\theta\hat{w}'$ is also in the feasible set for any positive scalar θ. Note that the function $\Psi\left(\beta, w\right)$ is linear in w and thus $\Psi\left(\beta^*, \theta\hat{w}'\right) = \theta \cdot \Psi\left(\beta^*, \hat{w}'\right)$. Due to the fact that this property is valid only in the case of the extreme ray, $\Psi\left(\beta^*, \hat{w}'\right) \ge 0$ implies that the objective value of problem (17) with β^* is unbounded. This contradicts the fact that t^* should be finite in (17) with β^*. Therefore, β^* cannot violate the feasibility cut (27).

Therefore, the feasible set, cut by new constraints, still contains the optimal solution (α^*, β^*). ■

According to Theorem 2, we can confirm that the problem in (18) of the RAS Algorithm can be replaced by (25) without compromising the optimality of the solution. Since we have relaxed the integer constraints, the optimal solution of (25), i.e., $\hat{\beta}$, may not be a fully integer matrix as it contains some decimals. According to the computational analysis in [33], Feasible Pump [32]–[34] is effective to find the feasible solution of the hard 0-1 MIP problem. Therefore, to obtain an integer matrix for calculating the lower and upper bounds of the problem $\Im_2$ more correctly, we adopt the Feasible Pump method to solve this problem by searching a nearest integer matrix with $\hat{\beta}$.

In the kth (excluding 0) iteration, we denote the decimal solution obtained from solving the problem in (25) by $\hat{\beta}^{(k)}$, and the integer solution by $\bar{\beta}^{(k)}$. The implementation of the Feasible Pump method to transform the decimal solution into the integer solution is described as follows.

At each iteration, the optimal decimal solution $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ are obtained by solving the integer-relaxed master problem in (25). Then, the rounding $\bar{\beta}^{(k)}$ of the given $\hat{\beta}^{(k)}$ can be computed by setting $\bar{\beta}_{ij}^{(k)} := \left[\hat{\beta}_{ij}^{(k)}\right]$ when $i \in N, j \in M$, if $\bar{\beta}_{ij}^{(k)} \ne \hat{\beta}_{ij}^{(k)}$, where $[\cdot]$ represents scalar rounding to the nearest integer. The (L_1-norm) distance between the optimal solution of (25) $\hat{\beta}^{(k)}$ and a given integer $\bar{\beta}^{(k)}$ is defined as follows:

$$\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}\right) = \sum_{i \in N, j \in M} \left|\hat{\beta}_{ij}^{(k)} - \bar{\beta}_{ij}^{(k)}\right|. \tag{28}$$

For a given integer $\bar{\beta}^{(k)}$, the distance can be rewritten as follows:

$$
\begin{aligned}
\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}\right) &= \sum_{0 \le \bar{\beta}_{ij}^{(k)} \le 1} D_{ij} \\
&= \sum_{\bar{\beta}_{ij}^{(k)} = 0} \left(\hat{\beta}_{ij}^{(k)} - 0\right) + \sum_{\bar{\beta}_{ij}^{(k)} = 1} \left(1 - \hat{\beta}_{ij}^{(k)}\right),
\end{aligned} \tag{29}
$$

based on the variable $D_{ij} = \left|\hat{\beta}_{ij}^{(k)} - \bar{\beta}_{ij}^{(k)}\right|$, we can obtain the double constraints, i.e.,

$$D_{ij} \ge \hat{\beta}_{ij}^{(k)} - \bar{\beta}_{ij}^{(k)} \text{ and } D_{ij} \ge \bar{\beta}_{ij}^{(k)} - \hat{\beta}_{ij}^{(k)}. \tag{30}$$

We then define the distance between $\hat{t}^{(k)}$ and $\bar{t}^{(k)}$ as follows:

$$\Delta\left(\hat{t}^{(k)}, \bar{t}^{(k)}\right) = \left|\hat{t}^{(k)} - \bar{t}^{(k)}\right|, \tag{31}$$

where $\bar{t}^{(k)}$ is computed with the constraints $\Psi\left(\beta, w^{(p)}\right) \le t, \forall p = 1, \ldots, k_1$ of (25) with the rounding integer $\bar{\beta}^{(k)}$. Similarly, we add a variable $D_t = |\hat{t}^{(k)} - \bar{t}^{(k)}|$ to construct another pair of constraints as follows:

$$D_t \ge \hat{t}^{(k)} - \bar{t}^{(k)} \text{ and } D_t \ge \bar{t}^{(k)} - \hat{t}^{(k)}. \tag{32}$$

Moreover, the solution $\hat{\beta}^{(k)}$ closest to $\bar{\beta}^{(k)}$ can be easily obtained by solving the following LP problem:

$$
\begin{aligned}
\min_{\hat{\beta}^{(k)}, \hat{t}^{(k)}} \quad & \Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right) \\
= \min_{\hat{\beta}^{(k)}, \hat{t}^{(k)}} \quad & \Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}\right) + \Delta\left(\hat{t}^{(k)}, \bar{t}^{(k)}\right) \\
\text{s.t.} \quad & \text{constraints of (25)}, \\
& D_{ij} \ge \hat{\beta}_{ij}^{(k)} - \bar{\beta}_{ij}^{(k)}, \\
& D_{ij} \ge \bar{\beta}_{ij}^{(k)} - \hat{\beta}_{ij}^{(k)}, \\
& D_t \ge \hat{t}^{(k)} - \bar{t}^{(k)}, \\
& D_t \ge \bar{t}^{(k)} - \hat{t}^{(k)}.
\end{aligned} \tag{33}
$$

If $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right) = 0$, which means that $\left(\hat{\beta}_{ij}^{(k)}, \hat{t}^{(k)}\right)$ is equal to $\left(\bar{\beta}_{ij}^{(k)}, \bar{t}^{(k)}\right)$, $\forall i \in N, \forall j \in M$. Therefore, $\left(\hat{\beta}_{ij}^{(k)}, \hat{t}^{(k)}\right)$ is a feasible MIP solution. Conversely, given a solution $\hat{\beta}^{(k)}$, the integer solution $\bar{\beta}^{(k)}$ close to $\hat{\beta}^{(k)}$ can be determined through rounding $\hat{\beta}^{(k)}$. In the Fast-RAS algorithm, the Feasible Pump method iterates with two pairs of solutions $\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}\right)$, where $\bar{\beta}^{(k)}$ is an integer matrix, and $\left(\hat{t}^{(k)}, \bar{t}^{(k)}\right)$, which are iteratively updated with the aim of reducing the distance $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$.

Specifically, initializing the rounding of $\hat{\beta}^{(k)}$ as an integer $\bar{\beta}^{(k)}$ (typically infeasible), the iteration can begin with any $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$, which is obtained by solving the problem in (25). At each pumping cycle, as $\left(\bar{\beta}^{(k)}, \bar{t}^{(k)}\right)$ is fixed, we try to find the solution $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ as close to $\left(\bar{\beta}^{(k)}, \bar{t}^{(k)}\right)$ as possible through the LP problem. If $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right) = 0$, then $\left(\bar{\beta}^{(k)}, \bar{t}^{(k)}\right)$ is a feasible solution of the MIP problem, and the algorithm stops. Otherwise, $\bar{\beta}^{(k)}$ is replaced by rounding of $\hat{\beta}^{(k)}$ to reduce $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$ for the future iteration.

The proposed process may prematurely be terminated due to the stalling issue, which happens that $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$ does not reduce anymore when updating $\bar{\beta}^{(k)}$ through rounding of $\hat{\beta}^{(k)}$. It means that each rounding element of $\bar{\beta}^{(k)}$ will not be changed compared with that in the last cycle. In this situation, part of $\bar{\beta}_{ij}^{(k)}$ will be modified heuristically by the Feasible Pump method, even though it will increase the current value of $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$.

THEOREM *3:* The Fast-RAS algorithm can effectively obtain an integer optimal solution compared with the RAS algorithm at each iteration of the Benders' Decomposition, approximately even equally.

Proof: At each iteration of the Benders' Decomposition process, we solve the integer-relaxed master problem (25) to obtain an optimal solution $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$. Although the integer relaxation extends the solution domain of the problem in (25), the MINLP problem can be transformed into the NLP problem, which can reduce computational complexity significantly. Let U and V be the feasible solutions of problem (18) and (25), according to Theorem 2, U is a subset of V, i.e., $U \subset V$. Therefore, there are two cases for solutions. In the first case, $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ is in the feasible set of the problem in (18), $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right) \in U$. In the second case, $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ is out of the the feasible set of the problem in (18), $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right) \in V, \notin U$. The proof for the two cases is given as follows.

In the first case, $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right) \in U$, $\hat{\beta}^{(k)}$ is a completely integer matrix, and all of its elements are binary integers, which means that $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ is also a feasible solution of the problem in (18). Moreover, as the problems in (18) and (25) have the same constraints, $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ is also the optimal solution of the problem in (18), which satisfies the following relationship:

$$f\left(\hat{\beta}^{(k)}\right) + \hat{t}^{(k)} = f\left(\beta^{(k)}\right) + t^{\dagger}, \tag{34}$$

where $\left(\beta^{(k)}, t^{\dagger}\right)$ is the optimal solution of the original master problem in (18). The solution $\hat{\beta}^{(k)}$ and $\hat{t}$ can be used to calculate the lower bound directly without any further processing.

In the second case, $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right) \in V, \notin U$, $\hat{\beta}^{(k)}$ is a decimal matrix and some of its elements are decimals. This means that the optimal solution $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ is infeasible to the problem in (18), and the relationship can be inferred as follows:

$$f\left(\hat{\beta}^{(k)}\right) + \hat{t}^{(k)} < f\left(\beta^{(k)}\right) + t^{\dagger}, \tag{35}$$

where $\beta^{(k)}$ is the nearest integer matrix to $\hat{\beta}^{(k)}$. At each cycle of Feasible Pump, we constantly slight the $\hat{\beta}^{(k)}$ to $\beta^{(k)}$ by solving the problem $min\ \Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$. Since the constraints of (25) are added into $min\ \Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$, this makes the solution $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ of each iteration always be a feasible solution of the problem in (25). When $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right) = 0$, $\hat{\beta}^{(k)}$ becomes an integer matrix and satisfies the following relationship:

$$f\left(\hat{\beta}^{(k)}\right) + \hat{t}^{(k)} \leq f(\beta) + t, \forall (\beta, t) \in V. \tag{36}$$

Therefore, $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right)$ converges to $\left(\hat{\beta}^{(k)}, t^{\dagger}\right)$, and the Feasible Pump method stops.

In order to avoid falling into a ineffective cycle, if the new $\bar{\beta}^{(k)}$ is the same as the old one and $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$ is not already zero, we introduce a variable D_{ij} into the Fast-RAS algorithm. At step 16, the number T_T of $\bar{\beta}_{ij}^{(k)}$ will provide a shift with the maximum D_{ij}, where T_T is a random integer in the range of $(T/2, 3T/2)$ and T is a given parameter. With this approach, we can find an approximate optimal solution within finite number of iterations. ■

The whole process of the Fast-RAS algorithm is described from step 1 to step 32, and the do-while iteration will terminate when the gap between the lower bound and upper bound is less than τ. Furthermore, from step 2 to step 22, we solve the master problem to obtain the lower bound through relaxing the integer constraints. At step 24, we calculate the optimal solution of the subproblem to obtain the upper bound of the objective function. Then, from step 25 to step 30, we estimate the boundedness of the subproblem to obtain the constraint added to the master problem. Specifically, the Feasibility Pump method starts at step 6 to step 21. In each pumping cycle at step 8, there is a pair of solutions $\left(\hat{\beta}_{ij}^{(k)}, \hat{t}^{(k)}\right)$ and $\left(\bar{\beta}_{ij}^{(k)}, \bar{t}^{(k)}\right)$. The first one is a feasible solution of (25) but may be decimal, and the another one is an integer matrix but may be infeasible. We try to reduce the distance $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$ at each iteration until it becomes 0. Here, $\left(\hat{\beta}_{ij}^{(k)}, \hat{t}^{(k)}\right)$ is a feasible integer solution of (25). From step 16 to step 18, a heuristic method is introduced to jump out the indefinite cycle.

Lemma 3 (Time complexity): The time complexity of the RAS algorithm is $O\left(\frac{1}{\tau}\left((nm)^2(nm+1)^3 + n^3m^3\right)\ln\left(\frac{1}{\varepsilon}\right)\right)$, and that of the Fast-RAS algorithm is $O\left(\frac{1}{\tau}\left(2^{nm} + n^3m^3\ln\left(\frac{1}{\varepsilon}\right)\right)\right)$. Compared with the RAS algorithm, the Fast-RAS algorithm can reduce the time complexity significantly.

Proof: Based on the detailed analysis in Appendix A, we prove that the time complexity of the Benders' Decomposition process is reduced from an exponential-time to a polynomial-time by the Fast-RAS algorithm. ■

VI. PERFORMANCE EVALUATION

In this section, we conduct simulations to compare the performance of the RAS and Fast-RAS algorithms from three

Algorithm 2 The Fast-RAS algorithm

Initialization: $k = 0, UB^{(0)} = +\infty, LB^{(0)} = 0, t = 0$.
while $UB^{(k)} - LB^{(k)} > \tau$ **do**
  if k=0 **then**
    Set $\beta^{(0)}$ and $C^{(0)}$ according to the network status.
  else
    Solve the master problem in (25) to obtain the optimal solution $\hat{\beta}^{(k)}$, $\hat{t}^{(k)}$.
    $\bar{\beta}^{(k)} := \left[\hat{\beta}^{(k)}\right]$ (rounding of $\hat{\beta}^{(k)}$). $Iter := 0$.
    Compute $\bar{t}^{(k)}$ with the constraints $\Psi\left(\beta, w^{(p)}\right) \leq t, \forall p = 1, \ldots, k_1$ and the rounding integer $\bar{\beta}^{(k)}$.
    while $(\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right) > 0$ & $Iter < maxI)$ **do**
      $Iter \leftarrow Iter + 1$.
      $\left(\hat{\beta}^{(k)}, \hat{t}^{(k)}\right) \leftarrow argmin\ \Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right)$.
      if $\Delta\left(\hat{\beta}^{(k)}, \bar{\beta}^{(k)}, \hat{t}^{(k)}, \bar{t}^{(k)}\right) > 0$ **then**
        if $\left[\hat{\beta}_{ij}^{(k)}\right] \neq \bar{\beta}_{ij}^{(k)}$ for at least one $i \in N, j \in M$ **then**
          Update $\bar{\beta}^{(k)} \leftarrow \left[\hat{\beta}^{(k)}\right]$.
        else
          For each $i \in N, j \in M$, define $D_{ij} := \left|\hat{\beta}_{ij}^{(k)} - \bar{\beta}_{ij}^{(k)}\right|$.
          Flip $T_T = rand\,(T/2, 3T/2)$ elements $\bar{\beta}_{ij}^{(k)}$ with the largest D_{ij}.
          Restart the cycle with new $\left(\bar{\beta}_{ij}^{(k)}, \bar{t}^{(k)}\right)$.
        end if
      end if
    end while
    Obtain the optimal integer solution $\beta^{(k)}$ and value $t^{\dagger}$, and then calculate the lower bound $LB^{(k)}$ with $\beta^{(k)}$.
  end if
  Solve the subproblem in (17) with $\beta^{(k)}$ to obtain the upper bound $UB^{(k)}$ as $\min\left\{f\left(\beta^{(k)} + U^{(k)}\right)\right\}$ and the optimal solution $w^{(k)}$.
  Let $t^*(\beta) = \Psi\left(\beta, w^{(k)}\right)$ to obtain the upper bound of t as t^*_{max} and the lower bound of t as t^*_{min}.
  if $t^*_{min} \leq t^{\dagger} \leq t^*_{max}$ **then**
    It is bounded, and we add a constraint $\Psi\left(\beta, w^{(k_1)}\right) \leq t$ to the master problem in (25).
  else
    It is unbounded, and we add a constraint $\Psi\left(\beta, w^{(k_2)}\right) \leq 0$ to the master problem (25).
  end if
  $k \leftarrow k + 1$.
end while
Select the optimal selected APs to switch to sleep mode according to AP's active matrix α through solving the primal subproblem in (16) with $\beta^{(k)}$. Return $\left(\alpha, \beta^{(k)}\right)$ as the optimal solution of the problem $\Im_2$.

aspects: computational complexity, converge condition and energy efficiency, i.e., energy saving. The simulation results demonstrate that the subset of the optimal solutions of the RAS algorithm can be obtained through the Fast-RAS algorithm, and the computational complexity can be reduced significantly. Moreover, we conduct experiments to validate the efficiency of the Fast-RAS algorithm compared with two classical energy-saving schemes, i.e., green-clustering algorithm [4] and cooperative energy-efficient method [12]. The experiment results show that the Fast-RAS algorithm can improve energy efficiency and network coverage than the other two methods.

All of the simulations are run with MATLAB R2014b on an Ubuntu 14.04 server with 32G memory and 2.2GHz Intel Xeon E5-4607 CPU * 2. Note that we solve the MINLP master problem by the SCIP [35] optimization toolbox which uses the branch-cut-and-price method. In the experiments, we set the error tolerance τ be 10^{-4} and the parameter T be 10.

A. Performance Comparison between RAS and Fast-RAS Algorithms

To compare the performance of the RAS and Fast-RAS algorithms, we choose a set of network topologies, where 12 APs are regularly deployed and a number of users are randomly placed in a 100m-by-100m area. The effective coverage radius of the APs is set to 40m, the throughput demand of each user is set randomly from a range between 2Mbps to 4Mbps, the AP transmission power is set to 30dBm, and a classic coloring algorithm is adopted to minimize the interference between the adjacent channels.

1) Computational Complexity: To validate the effectiveness of the proposed algorithms under different network loads, we vary the number of users from 20 to 100 with a step of 20. As the users are randomly placed in the area, we repeat the simulations 50 times with different distributions of users at each number to obtain accurate results. The computational time against the number of users of the RAS and Fast-RAS algorithms are depicted in Fig. 3. We observe that when the number of users increases from 20 to 100, the computational time of the RAS algorithm increases sharply compared with that of the Fast-RAS algorithm. Especially, with 100 users, the average computational time of the RAS algorithm is 2.5 times higher than that of the Fast-RAS algorithm.

It is clear that the Fast-RAS algorithm can effectively reduce the computational time compared with the RAS algorithm under the same condition. Moreover, regarding the Benders' Decomposition employed in the RAS and Fast-RAS algorithms, the numbers of iterations with different numbers of users are depicted in Fig. 4. We find that the number of iterations used by the Fast-RAS algorithm is always higher than that of the RAS algorithm in each experiment. However, the difference is relatively small. The reason is that the Fast-RAS algorithm uses a heuristic algorithm to obtain a sub-optimal solution in each iteration, which enhances the rate of convergence slightly.

Furthermore, we compare the complexity of our algorithms with the other two heuristic energy-saving algorithms in the same scenario. As shown in Fig. 3, the heuristic algorithms have lower computational time as expected. By contrast, our algorithms taking the user's QoE and network status into account incur more time complexity. Moreover, to find a solution closer to the optimal solution, the Fast-RAS algorithm still needs more iterations to satisfy the condition $UB - LB < \tau$.

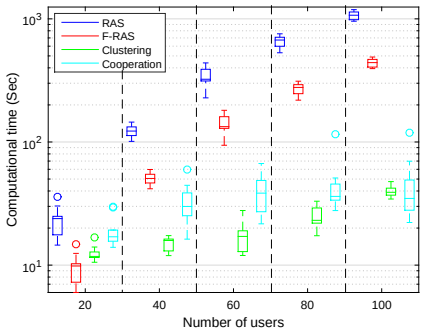


Fig. 3: Range of computational time vs. number of users.

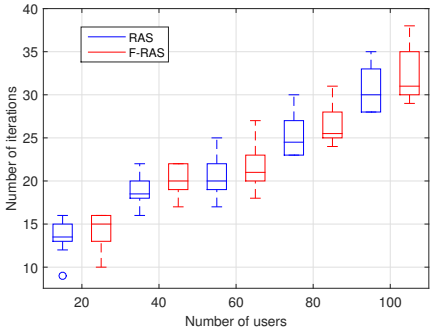
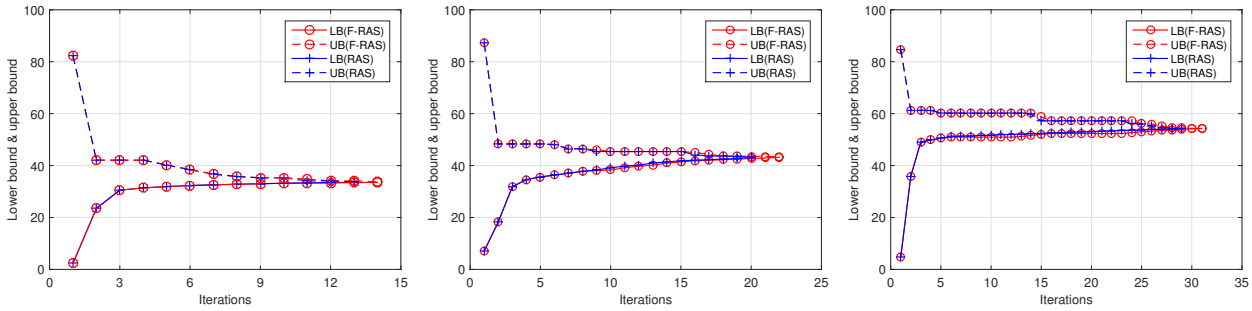
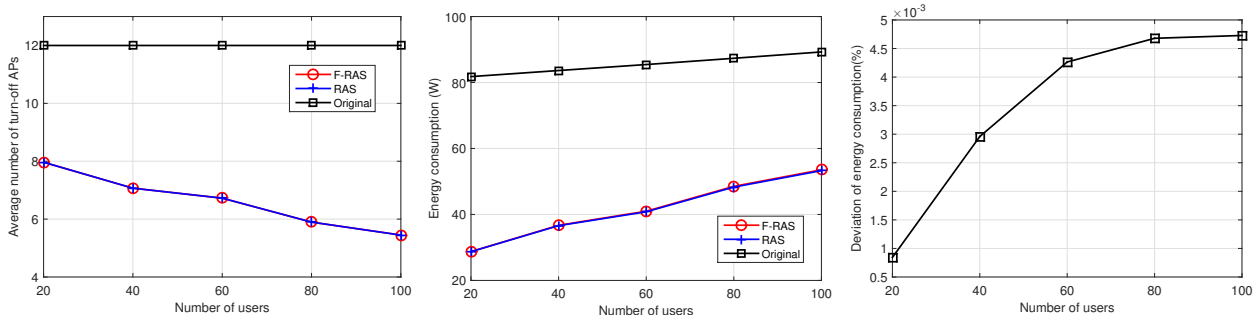


Fig. 4: Range of computational iteration vs. number of users.



(a) Lower bound & upper bound with 20 users. (b) Lower bound & upper bound with 60 users. (c) Lower bound & upper bound with 100 users.

Fig. 5: The converge condition of RAS and Fast-RAS algorithm with the number of users are 20, 60 and 100.



(a) The average number of turn-off APs. (b) System energy consumption. (c) Error of energy consumption.

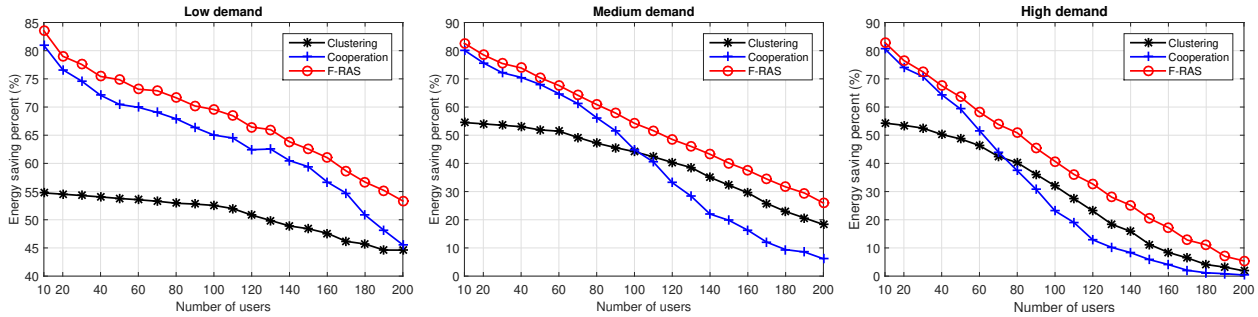
Fig. 6: The number of turn-off APs and energy consumption by RAS and Fast-RAS with different number of users.

In the simulation, we set τ to a small value of 10^{-4}. Actually, in practical applications, the convergence condition of the Fast-RAS algorithm can be relaxed by increasing τ (such as 10^{-2} or larger), which can drastically reduce the number of iterations and enhance the practicability of the algorithm.

2) Convergence Condition: To evaluate the convergence of the RAS and Fast-RAS algorithms, we vary the number of users from 20 and 100 and observe the changes of upper bound and lower bound. The convergence conditions are depicted in Fig. 5. At the early iterations, as the Fast-RAS algorithm only finds a subset of the optimal solution, there is a wider gap between the upper bound and the lower bound than that of the RAS algorithm. When iterations proceed, the difference becomes smaller gradually. At the final iteration, the Fast-RAS algorithm can obtain the same minimum value of the objective function as that of the RAS algorithm and has only 0.13%, 0.39% and 0.46% deviations with 20, 60 and 100 users, respectively. It is clear that the Fast-RAS algorithm can achieve the suboptimal solution close to the optimal solution obtained by the RAS algorithm.

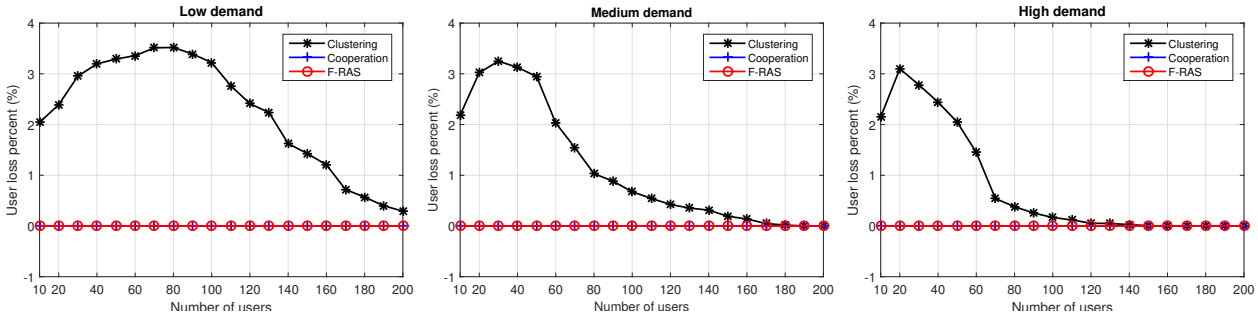
3) Performance Comparison: Based on the same network setting, we conduct experiments to compare the energy efficiency of WLAN systems using the RAS and Fast-RAS algorithms. We repeat the experiments 50 times to obtain the average values of the results, which are shown in Fig. 6.

The average number of APs that are selected to be turned off and the energy consumption under the different number of users are shown in Fig. 6 (a) and Fig. 6 (b), respectively. When the number of users increases, the number of turned-off APs decreases and the minimum energy consumption increases.



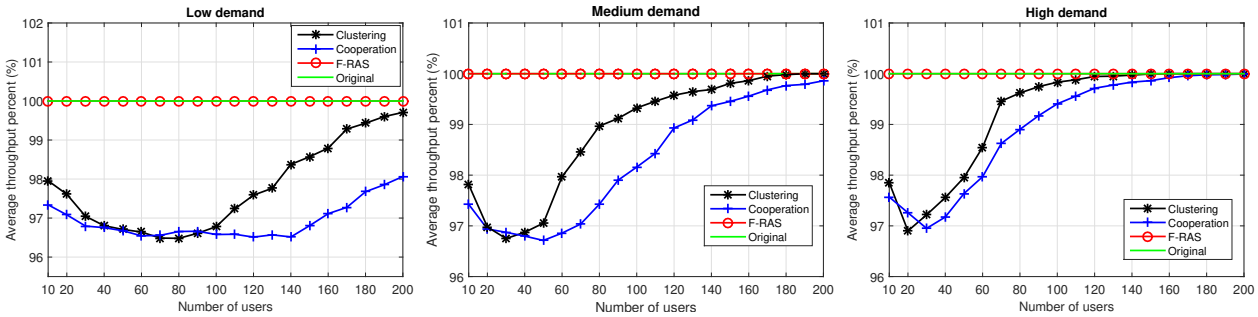
(a) Low user's throughput demand (0-4Mbps). (b) Medium throughput demand (0-8Mbps). (c) High throughput demand (0-12Mbps).

Fig. 7: Energy saving with low, medium and high users' throughput demands by three energy-saving algorithms.



(a) Low user's throughput demand (0-4Mbps). (b) Medium throughput demand (0-8Mbps). (c) High throughput demand (0-12Mbps).

Fig. 8: User loss with low, medium and high users' throughput demands by three energy-saving algorithms.



(a) Low throughput demand (0-4Mbps). (b) Medium throughput demand (0-8Mbps). (c) High throughput demand (0-12Mbps).

Fig. 9: Average throughput with low, medium and high users' throughput demands by three energy-saving algorithms.

Since the Fast-RAS algorithm can obtain the suboptimal solution close to the optimal solution, the number of turned-off APs and the energy efficiency are nearly the same as that of the RAS algorithm. Moreover, we quantify the deviation of energy consumption between the RAS and Fast-RAS algorithms. As shown in Fig. 6 (c), the deviation is small and varies from 0.084% to 0.473%, which is due to the selection of user re-association to new APs. Nonetheless, the deviation is in acceptable error tolerance.

B. Performance Comparison with Classical Algorithms

Next, we evaluate the energy efficiency of the Fast-RAS algorithm compared with the green-clustering algorithm [4] and cooperative energy-efficient method [12] in three aspects, i.e., energy saving, effective coverage and user's throughput. To simulate a real and comprehensive network condition, we expand the experiment area to 100m-by-100m, set the number of APs with regularly deployment to 20 and use three levels of user's throughput demand with normal distribution from 0-4Mbps, 0-8Mbps to 0-12Mbps. Moreover, we adopt three network topologies with different AP density and network load, and vary the number of users from 10 to 200 to test the performance.

1) Energy efficiency: As shown in Fig. 7, the Fast-RAS algorithm has the better energy efficiency in all settings. When there are only 10 users with low user throughput demand, the Fast-RAS algorithm obtains the high energy-saving rate which

is close to that of the cooperative method. The rate is almost 30 percent higher than that of the green-clustering algorithm. As the number of users increases, to satisfy the user throughput demand, the rate of energy saving reduces gradually. When the number of users increases to 200, the energy-saving rates of both the green-clustering algorithm and cooperative method decrease to 45%, but that of the Fast-RAS algorithm is still maintained at 53%.

The same phenomenon can be observed in medium user throughput demand. When the number of users increases to 200, the energy-saving rate of the Fast-RAS algorithm is 28%, which is more stable than that of the green-clustering algorithm with 20% and cooperative method with 7%. Moreover, when the user throughput demand increases nearly to the total capacity, to guarantee the user QoS requirement, it is difficult to select APs to turn off to save energy. Nonetheless, since the Fast-RAS algorithm can optimize user re-association, when the number of users increases to 200, the energy-saving rate of Fast-RAS algorithm is still around 5%, which is higher than that of the green-clustering algorithm and cooperative method.

Evidently, for low or high user throughput demands, the Fast-RAS algorithm always achieves the highest energy-saving rate. The reason is that the Fast-RAS algorithm provides more effective user re-association scheduling to choose many APs to turn off. Therefore, the Fast-RAS algorithm has 3% to 30% increase of energy-saving rate compared with other two classical algorithms.

2) Effective coverage: As shown in Fig. 8, since the Fast-RAS and cooperative algorithms can satisfy absolutely effective coverage, the better user coverage can be achieved than that of the green-clustering algorithm in all experiment settings. The reason is that both the Fast-RAS and cooperative algorithms make user re-association as the basic guarantee to select APs to turn-off. However, the green-clustering algorithm divides the APs into several clusters with one or more cluster-head AP, and other normal APs are selected to be turned off to save energy. This algorithm thus cannot ensure that all users can be re-associated with its cluster-head AP in practice. Consequently, it inevitably leads to user coverage loss.

3) Influence on user's throughput: As shown in Fig. 9, as user throughput guarantee is considered in our energy efficiency model, in all user throughput demand levels the average user throughput achieved by the Fast-RAS algorithm is stable around the original throughput, which is 3.4% higher than the best throughput obtained from the green-clustering and cooperative algorithms.

VII. Conclusion

In this paper, we have proposed the concept to aggregate and associate WLAN users to fewer APs so that more APs can be turned off to reduce energy consumption. This is achievable without sacrificing the demands of users' QoE and network coverage. Specifically, we have formulated the energy efficiency problem as an MINLP problem as well as transformed and solved it by using the RAS algorithm which is based on classical Benders' Decomposition method. To further reduce the computational complexity, we have proposed the Fast-RAS algorithm to relax the binary integer constraints to transform the MINLP problem into an NLP problem, which can be solved by using the Feasible Pump method, namely, the Fast-RAS algorithm.

The performance of the RAS and Fast-RAS algorithms has been validated via extensive simulations. The results demonstrate that both the RAS and Fast-RAS algorithms converge to globally optimal solution within reasonable time period. Moreover, compared with existed methods, the Fast-RAS algorithm can achieve up to 20% improvement of energy saving while maintaining effective network coverage and user's QoE.

Appendix A
Proof of Time Complexity

Let K_m and K_s represent the time complexity of the master problem and sub-problem. For the estimation of K_s, which are identical in the RAS and Fast-RAS algorithms, we define it as the LP problem and solve it by using the Primal-Dual Interior Point method. According to [36], the LP problem converges to an ε-accurate solution with the worst case bound of $O\left(\sqrt{x}\ln\left(1/\varepsilon\right)\right)$ in terms of the number of iterations. Here, each iteration requires $O\left(x^{2.5}\right)$ arithmetic operations, where x is the number of variables in the LP problem. Thus, the value of K_s is estimated as follows:

$$K_s = O\left(n^3 m^3 \ln\left(\frac{1}{\varepsilon}\right)\right), \tag{37}$$

where $n*m$ is the length of vector w in the subproblem and also represents the number of constraints, and ε is the convergence accuracy which is set to a sufficiently small value, e.g., 10^{-4}.

Solving the master problem is the major difference between the RAS and Fast-RAS algorithms in terms of the incurred computational complexity. The Branch and Bound algorithm (BB) is a classical and popular method to solve an MINLP problem [37]. It can be used to solve the RAS master problem. In our problem, there are n APs and m users, and $\beta^{(k)}$ is a 0/1 matrix, so it increases to $(2^{nm}-1)$ nodes. The time complexity is K_{m-R} is $O\left(2^{nm}\right)$, and the total number of arithmetic operations needed at each Bender iteration of the RAS algorithm K_R is defined as follows:

$$K_R = K_{m-R} + K_s = O\left(2^{nm} + n^3 m^3 \ln\left(\frac{1}{\varepsilon}\right)\right). \tag{38}$$

In the Fast-RAS algorithm, the feasibility pump cycle is introduced to replace the master problem. At each iteration, compared with other operations such as rounding and traversal, step 11 incurs the largest complexity while it is only the LP problem. Therefore, the number of arithmetic operations is easily estimated as $O\left((nm+1)^3 \ln\left(1/\varepsilon\right)\right)$, where $n*m+1$ is the size of $\hat{\beta}^{(k)}, \hat{t}^{(k)}$. On the other hand, according to the complexity analysis for the Feasibility Pump method [32], [33], a polynomial-time $O\left((nm)^2\right)$ can describe the complexity of our binary case. As such, the time complexity K_{m-FR} is

$$K_{m-FRAS} = O\left((nm)^2 (nm+1)^3 \ln\left(\frac{1}{\varepsilon}\right)\right). \tag{39}$$

Similarly, the total number of arithmetic operations needed at each Bender iteration of the Fast-RAS algorithm is defined:

$$K_{FR} = K_{m-FR} + K_s = O\left((nm)^2(nm+1)^3 \ln\left(\frac{1}{\varepsilon}\right) + n^3m^3 \ln\left(\frac{1}{\varepsilon}\right)\right). \quad (40)$$

Based on the previous analysis, we can prove that the time complexity of each Bender iteration is reduced from exponential-time to polynomial-time in the Fast-RAS algorithm. Therefore, the number of Bender iterations H, which cannot reach exponential-level, plays a weak role for complexity comparison between the RAS and Fast-RAS algorithms. We assume that H is $O(1/\tau)$, which is highly dependent on the accuracy threshold τ between upper and lower bounds. Thus, the total time complexities of the RAS and Fast-RAS algorithms are estimated as follows:

$$TC_R = H \cdot K_R = O\left(\frac{1}{\tau}\left(2^{nm} + n^3m^3 \ln\left(\frac{1}{\varepsilon}\right)\right)\right), \quad (41)$$

$$TC_{FR} = H \cdot K_{FR} = O\left(\frac{1}{\tau}\left((nm)^2(nm+1)^3 + n^3m^3\right) \ln\left(\frac{1}{\varepsilon}\right)\right). \quad (42)$$

It is worth noting that as a heuristic method, the Feasibility Pump method may not always be able to achieve an optimal solution as the BB algorithm at each Bender iteration. This results in more Bender iterations in the Fast-RAS algorithm than that of the RAS algorithm. Our experiments also confirm this result. Nonetheless, the errors are small and do not affect the performance of the algorithm considerably. Therefore, we can conclude that the Fast-RAS algorithm can improve the time complexity compared with the RAS algorithm significantly.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (NSFC) under grant 61402065 and the Key Technology R&D Project of Chongqing Technology Innovation and Application Demonstration under grant (cstc2018jszx-cyzdX0120).

REFERENCES

[1] Z. Zhu, P. Lu, J. Rodrigues, and Y. Wen, "Energy-Efficient Wideband Cable Access Networks in Future Smart Cities," *IEEE Communications Magazine*, vol. 14, no. 6, pp. 798–814, 2009.

[2] Cisco, "Cisco Visual Networking Index: Forecast and Trends, 2017-2022 White Paper," February 27. 2019.

[3] E. Goma, *et al.*, "Insomnia in the access: or how to curb access network related energy consumption," *ACM SIGCOMM Computer Communication Review*, vol. 41, pp. 338–349, 2011.

[4] A. P. Jardosh *et al.*, "Green wlans: on-demand wlan infrastructures," *Mobile Networks and Applications*, vol. 14, no. 6, pp. 798–814, 2009.

[5] Gomez. K, Boru. D, Riggio. R and et al. "Measurement-based modelling of power consumption at wireless access network gateways,". *Computer Networks*, vol. 56, no. 10, pp. 2506–2521, 2012.

[6] A. P. Jardosh, *et al.*, "Towards an energy-star WLAN infrastructure," in *Proceedings of 8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile)*, pp. 85–90, 2007.

[7] K. Gomez, R. Riggio, T. Rasheed and G. Granelli, "Analysing the energy consumption behavior of WiFi networks," in *Proceedings of IEEE Online Conference on Green Communications*, pp. 98–104, 2011.

[8] Ł. Budzisz, *et al.*, "Dynamic resource provisioning for energy efficiency in wireless access networks: A survey and an outlook," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2259–2285, 2014.

[9] F. G. Debele, *et al.*, "Designing resource-on-demand strategies for dense wlans," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2494–2509, 2015.

[10] T. Tanaka, *et al.*, "Automatic and cooperative sleep control strategies for power-saving in radio-on-demand wlans," in *Proceedings of IEEE Green Technologies Conference*, pp. 293–300, 2013.

[11] K. Gomez, *et al.*, "Morfeo: Saving energy in wireless access infrastructures," in *Proceedings of IEEE WoWMoM*, pp. 1–6, 2013.

[12] C. Rossi, *et al.*, "Cooperative energy-efficient management of federated wifi networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, pp. 2201–2215, 2015.

[13] A. P. C. D. Silva, *et al.*, "Energy-performance trade-off in dense WLANs: A queuing study," *Computer Networks*, vol. 56, no. 10, pp. 2522–2537, 2012.

[14] G. Bhalla, *et al.*, "CrowdAP: Crowdsourcing driven AP coordination for improving energy efficiency in wireless access networks," in *Proceedings of IEEE International Conference on Communications (ICC)*, pp. 1–6, 2016.

[15] R. Riggio *et al.*, "Programming Abstractions for Software-Defined Wireless Networks," *IEEE Transactions on Network & Service Management*, vol. 12, no. 2, pp. 146–162, 2015.

[16] C. Xu, *et al.*, "A Novel Multipath-Transmission Supported Software Defined Wireless Network Architecture," *IEEE Access*, vol. 5, pp. 2111–2125, 2017.

[17] A. Amokrane, R. Langar, R. Boutaba, and G. Pujolle, "Flow-Based Management For Energy Efficient Campus Networks," *IEEE Transactions on Network & Service Management*, vol. 12, no. 4, pp. 565–579, 2015.

[18] Y. J. Chen, Y. H. Shen, and L. C. Wang, "Achieving energy saving with QoS guarantee for WLAN using SDN," in *Proceedings of IEEE International Conference on Communications (ICC)*, pp. 1–7, 2016.

[19] N. Trabelsi, *et al.*, "User Association and Resource Allocation Optimization in LTE Cellular Networks," *IEEE Transactions on Network & Service Management*, vol. 14, no. 2, pp. 429–440, 2017.

[20] Q. Fan, H. Lu, P. Hong, and Z. Zhu, "Throughput-Power Tradeoff Association for User Equipments in WLAN/Cellular Integrated Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3462–3474, 2017.

[21] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[22] A. M. Geoffrion, "Generalized benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, 1972.

[23] A. Kheirkhah, H. Navidi and M. M. Bidgoli, "An Improved Benders Decomposition Algorithm for an Arc Interdiction Vehicle Routing Problem," *IEEE Transactions on Engineering Management*, vol. 63, no. 2, pp. 259–273, 2016.

[24] L. P. Qian, Y. J. A, Zhang, Y. Wu and J. Chen, "Joint base station association and power control via Benders' decomposition," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1651–1665, 2013.

[25] A. Nasri, S. J. Kazempour, S. J. Conejo and M. Ghandhari, "Network-constrained AC unit commitment under uncertainty: A benders' decomposition approach," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 412–422, 2016.

[26] N. McKeown, *et al.*, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[27] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Communications magazine*, vol. 47, no. 7, pp. 20–26, 2009.

[28] Sequeira. L, *et al.*, "Building an SDN enterprise WLAN based on virtual APs," *IEEE Communications Letters*, vol. 21, no. 2, pp. 374–377, 2017.

[29] Rangisetti. A. K, *et al.*, "Load-aware hand-offs in software defined wireless LANs," *International Conference on Wireless and Mobile Computing, NETWORKING and Communications*, pp. 685–690, IEEE, 2014.

[30] C. A. Floudas, A. Aggarwal and A. R. Ciric, "Global optimum search for nonconvex NLP and MINLP problems," *PComputers and Chemical Engineering*, vol. 13, no. 10, pp. 1117–1132, 1989.

[31] R. K. Martin, "Large scale linear and integer optimization: a unified approach," *Springer Science and Business Media*, 2012.

[32] M. Fischetti, F, Glover, and A. Lodi, "The feasibility pump," *Mathematical Programming*, vol. 104, no. 1, pp. 91–104, 2005.

[33] P. Bonami, G. Cornuejols, A. Lodi and F. Margot, "A feasibility pump for mixed integer nonlinear programs," *Mathematical Programming*, vol. 119, no. 2, pp. 331–352, 2009.

[34] M. L. Boland, *et al.*, "Boosting the feasibility pump," *Mathematical Programming Computation*, vol. 6, no. 3, pp. 255–279, 2014.
[35] S. Vigerske, and A. Gleixner, "SCIP: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework," *Optimization Methods and Software*, pp. 1–31, 2017.
[36] J. Renegar, "A polynomial-time algorithm, based on Newton's method, for linear programming," *Mathematical Programming*, vol. 40, no. 1, pp. 59–93, 1988.
[37] P. Bonami, and J. P. Goncalves "Heuristics for convex mixed integer nonlinear programs," *Computational Optimization and Applications*, vol. 51, no. 2, pp. 729–747, 2012.

Chuan Xu (M'16) received the B.E. and M.E. degrees in communication engineering from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2003 and 2006, respectively, and the Ph.D. degree in control theory and engineering from Chongqing University, China, in 2012. He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, China. His research interests are in the areas of wireless communication, software-defined networking, and Internet of Things.

Jiajie Wang is currently a researcher staff member in the Future Networks research institute at Chongqing University of Posts and Telecommunications, China. His research interests are on energy efficient wireless network, software-defined networking.

Zuqing Zhu (M'01-SM'12) received his Ph.D. degree from the Department of Electrical and Computer Engineering, University of California, Davis, in 2007. From 2007 to 2011, he worked in the Service Provider Technology Group of Cisco Systems, San Jose, California, as a Senior Engineer. In January 2011, he joined the University of Science and Technology of China, where he currently is a Full Professor. His current research interests are on optical networks, software/knowledge-defined networking (SDN/KDN), network function virtualization (NFV) and datacenter networks.

Dusit Niyato (M'09-SM'15-F'17) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of energy harvesting for wireless communication, Internet of Things, and sensor networks.