

Near-optimal user-cell association schemes for real-world networks

Yonglong Zhang, Dilip Bethanabhotla, Tianyu Hao and Konstantinos Psounis
University of Southern California

Abstract—The growing demand for wireless bandwidth makes WiFi deployments denser and pushes cellular networks to adopt a denser, small-cell architecture. In such dense environments, users have multiple options when it comes to selecting an access point (AP) to associate with the network, and the user-cell association scheme that is used has a large impact on overall network performance. Industry is currently using simplistic, sub-optimal approaches to select an AP for each user, while academia has produced optimal schemes under unrealistic assumptions, which prevents them from ever being used in practice. In this paper we design high-performance user-cell association schemes which can be deployed in real-world networks. The performance of the proposed schemes is shown to be near-optimal via both formal performance bounds under realistic assumptions, and simulation results under real-world setups.

I. INTRODUCTION

Modern wireless devices such as tablets and smartphones are pushing the demand for higher and higher wireless data rates while causing significant stress to existing wireless networks. While successive generations of wireless standards achieve continuous improvement, it is the general understanding of both academic research and industry that a significant increase in wireless traffic demand can be met only by a dramatically denser spectrum reuse, i.e., by deploying more base stations (BSs)/access points (APs) per square kilometer, coupled with advanced techniques to reduce inter-cell interference.

Enterprise WiFi networks have been deployed following this paradigm for years, see, for example, the particularly dense deployments in conference halls, university campus centers, stadiums, etc. Cellular networks, unable to satisfy the bandwidth demand of data plans, resort to WiFi offloading, i.e. they deploy WiFi networks to offload the cellular network. Future cellular network architectures (4G+ and most notably 5G) will be designed in a manner that avoids WiFi offloading, by densely deploying small cells (often referred to as pico cells, micro cells, femto cells, etc.) in addition to the traditional macro cells. In such dense environments, user terminals have multiple options when it comes to selecting a BS/AP to associate with the network. Even in the presence of multiple non-overlapping channels like in the case of WiFi, there are still multiple APs which may operate on the same channel and are within range of a nearby user.

The user-cell association scheme that may be used in such networks has a large impact on network performance. In existing enterprise WiFi networks, this selection is usually based on the received signal strength indication (RSSI) values that a user gets from nearby APs: Users simply associate with the AP with the strongest RSSI reading. Some manufacturers have adopted slightly more advanced schemes attempting

to impose some sort of load balancing [1]: If an AP has already too many users associated with it, it will reject further association requests forcing users to select another nearby AP. Despite their simplicity, these real world schemes waste precious system capacity and lead to suboptimal performance [2].

There is a large body of work in academia to optimally address the user-cell association problem under a large range of scenarios, see, for example, [3] and the references therein. However, these works resort to a number of unrealistic assumptions. For example, users may be allowed to connect to multiple BSs/APs and activate the connection with each of them for a fraction of time, users may be allowed to switch between BSs/APs as often as a new user arrives or an existing one departs, and users are assumed to be backlogged. Such assumptions make the corresponding approaches not immediately implementable.

To address the disconnect between theory and practice, in this paper we present user-cell association schemes which are provably near-optimal yet they are amenable to practical implementation. Specifically, under scenarios where users are not backlogged but instead have a target rate which depends on the application that they are using, we present a greedy algorithm which associates each user with a single AP and formally bound the performance with respect to the optimal user-cell association scheme. To better understand the effect of not allowing users to switch APs, we also present and formally bound the performance of a best-response algorithm which is allowed to switch users if this improves overall performance. Our formal results are based on algorithmic game theory concepts [4].

In addition to providing formal bounds on the performance of the proposed algorithms, we also conduct a number of realistic simulations to access the real-world gains from our schemes, as well as to study how close they perform to the optimal in practice. We show that our schemes can improve the performance of enterprise WiFi networks by a sizable amount over existing approaches (often exceeding 30%), and that they are reasonably close to the optimal (10-30% off the optimal depending on the scenario). Since the fundamental principles behind a user-cell association scheme are common in cellular and WiFi networks and in view of the fact that today the problem of user-cell association is acute only in the context of dense enterprise WiFi networks, in the rest of the paper we will refer to the infrastructure nodes (BSs/APs) as APs.

The outline of the rest of the paper is as follows. In Section II, we give a brief overview of prior work on the user-AP association problem and highlight the contributions of this paper. In Section III, we describe the system model and mathematically formulate the problem we wish to solve. We then describe

in Section IV two specific algorithms which attempt to solve the problem: 1) the *greedy algorithm* where the association decisions are made online as the users arrive and 2) the *best response algorithm* where the users are allowed to switch APs multiple times in order to receive better throughputs. In Section V, we provide formal bounds on the performance of the greedy and the best response algorithms in comparison to the optimal off-line solution. In Section VI, we perform extensive simulations of both the greedy and best response algorithms in realistic network setups and verify near-optimal performance. Finally, in Section VII, we discuss how these algorithms could be implemented as protocols in a practical setting.

II. RELATED WORK

The traditional RSSI based association can be arbitrarily suboptimal in both WLANs with a dense deployment of APs, like in an enterprise-WiFi network, and in cellular heterogeneous networks [2] based on nested tiers of smaller and smaller cells operating at higher and higher frequencies, which are deployed in order to target “hot-zones” [5]. In the context of cellular networks, “biasing” is a commonly proposed heuristic method for heterogeneous networks, where the RSSI is artificially scaled by a “bias” term that depends on the BS tier, to inherently steer the user association towards lower-tier BSs and therefore “off-load” the macro-cell (see [6] and the references therein). A similar approach popularly known as “cell breathing” is often proposed in the context of WLANs to off-load users from highly loaded APs to the lightly loaded ones by dynamically expanding or contracting the AP coverage area in order to balance the load evenly among all the APs in the networks [7], [8].

A more systematic approach to user-AP association as an optimization problem was taken in [3], [6], [9]–[12]. While summarizing this large body of literature is out of scope, we mention that [3] was the first to offer a distributed algorithm which was formally proved to be optimal. Specifically, a convex optimization problem was formulated where a network-wide objective function was maximized under fractional association constraints, and both centralized and distributed algorithms were proposed to solve the convex program optimally. The distributed algorithm was first shown to converge to a Nash equilibrium and any such Nash equilibrium was shown to be globally optimal in the sense of solving the originally formulated convex program. However, [3] assumes a static network where the locations of both the users and the APs remain fixed and there are no new users arriving into the system. Also, the distributed algorithm allows users to switch multiple times between different APs until the system reaches an equilibrium where no user is motivated to switch any longer. This is clearly not desirable in dynamic network conditions, since a user may have to continuously switch between different APs in order to be adapt to the dynamics.

In the process of obtaining formal performance bounds for our algorithms, we utilize ideas from prior work on on-line load balancing [4], [13]–[15], where bounds are given on the ratio of the performance of the online algorithm to that of the off-line optimal solution. For example, [4], [13] study the load balancing problem in restricted settings where jobs need to be assigned to *identical* or *constant speed* machines

which is not the case in this paper because different users have different peak rates to different APs depending on their locations. Our problem falls into the general class of problems popularly known as *scheduling jobs on unrelated machines* [14]–[16]. However, these works provide pessimistic bounds on the performance ratios under arbitrary machine speeds and job service requirements. In this work, we provide optimistic bounds on the performance ratios under the assumption that the user-AP peak rates (“machine”/AP speed when servicing the corresponding “job”/user) do not get arbitrarily small, as is the case in practice.

III. SYSTEM MODEL AND PROBLEM DEFINITION

We consider a system formed by a set of AP J serving a set I of several users distributed over a given coverage area. We use the index i to denote the user and j to denote the AP. Let C_{ij} denote the peak rate or capacity of the PHY layer link between user i and AP j . In addition, we assume that each user i has an *application-aware target rate* T_i . For instance, T_i would be of the order of Kbps if user i is browsing webpages. On the other hand, if user i were to stream video, then T_i would be of the order of Mbps.

Note that the peak rate C_{ij} is the throughput that user i would get if AP j were to serve user i alone on all time-frequency resources. However, when multiple users are associated to an AP, the time-frequency resources at the AP are shared among the associated users and therefore user i would get a throughput which is only a fraction α_{ij} of the peak rate C_{ij} depending on the scheduling policy employed by the AP j .

In this paper, we assume that each AP employs the same *target rate-aware proportional fairness* scheduling policy for sharing resources among the users associated with it. That is, each user i associated with AP j gets *throughput* $\alpha_{ij}C_{ij}$ where the fraction α_{ij} is given as:

$$\alpha_{ij} = \frac{\frac{T_i}{C_{ij}}}{\sum_{i' \in I(j)} \frac{T_{i'}}{C_{i'j}}} \quad (1)$$

with $I(j)$ denoting the set of users associated with AP j .

Note from the expression (1) that the policy favors users with higher target rates by allocating a larger fraction α_{ij} of resources.

We define the *satisfaction* s_i for a user i associated with AP j as the ratio of the throughput offered by the AP j and the target rate T_i , i.e.,

$$\begin{aligned} s_i &= \frac{\alpha_{ij}C_{ij}}{T_i} \\ &= \frac{1}{\sum_{i' \in I(j)} \frac{T_{i'}}{C_{i'j}}} \end{aligned} \quad (2)$$

Note that the term $\sum_{i' \in I(j)} \frac{T_{i'}}{C_{i'j}}$ in the denominator can be interpreted as the *load* at the AP j . Thus, user i 's satisfaction is the reciprocal of the *load*

$$l_j = \sum_{i' \in I(j)} \frac{T_{i'}}{C_{i'j}} \quad (3)$$

at the AP j with which it is associated.

We now wish to find the optimal association of users to APs such that the minimum of all the users' satisfaction ratios is maximized. However, from (2), since the user satisfaction is the reciprocal of the load at the AP to which it is associated, we can deduce that maximizing the minimum of all users' satisfaction ratios is equivalent to minimizing the maximum load among all the APs in the system. This is also known as *makespan* minimization in theoretical computer science literature where makespan is a term used for the maximum load across all the APs (see [4]).

However, finding an optimal association is an NP-hard problem with running time $O(|J|^{|I|})$ ($|J|$ and $|I|$ denote the number of APs and the number of users in the system respectively). In order to solve this problem tractably, we resort to near-optimal schemes which are practically implementable. In the next section, we describe two such association algorithms and in Section V, we analyze their performance.

IV. ALGORITHMS DESCRIPTION

In this section, we describe two specific algorithms 1) *the greedy algorithm* and 2) *the best response algorithm* which attempt to minimize the maximum load across the APs. We assume that the AP locations remain fixed. On the other hand, both algorithms are flexible to changing user population, i.e., users leaving or arriving into the system. Furthermore, they are also flexible to both distributed and centralized implementations.

A. The Greedy Algorithm

The greedy algorithm is simple to describe and it works as follows: When a new user i arrives into the system, it greedily (or selfishly) chooses the AP j^* with the minimum load, i.e.,

$$j^* = \arg \min_{j \in J} \left\{ l_j + \frac{T_i}{C_{ij}} \right\} \quad (4)$$

and associates with it. Once it assigns itself to j^* , it stays there forever. Note that the load expression in (4) also takes into account user i 's contribution to the load if it were to join j . Also, note that in a practical large-scale network scenario, e.g. an enterprise WiFi network, the set of APs J that user i would consider are the APs that are within range of the user rather than all the APs of the network. We say that these APs are in the *neighborhood* of a user.

The algorithm is *online* in nature because the user-AP association decision is made on-the-fly as the users arrive into the system. Once the association decision is made, the user is not allowed to switch to another AP in the future. In addition, the algorithm is distributed in nature since the association decision is made only by the newly arriving user. From (4), we observe that the user needs to learn the quantities l_j and C_{ij} from all the APs j in its neighborhood and then find the best BS j^* . This information can be easily read off from the beacon signals that the APs in the neighborhood of user i periodically transmit. This approach is popularly known as *beacon-stuffing* [17] and is widely used in practice for location-aware delivery of advertisements, coupons etc.

Furthermore, it is immediate to notice that the algorithm can also be implemented in a centralized manner where a centralized controller could order the users in some arbitrary way and make the association decision (4) for each user one after the other in that order.

B. The Best Response Algorithm

In this case, the same decision (4) is used to associate a user with a base station. However, what makes this algorithm different from the greedy algorithm is that a user is allowed to switch its association multiple times. Since the association decision (4) is a local and selfish decision taking into account only user i 's satisfaction, it is not necessarily the best global decision for all the users in the network. In particular, this user-centric selfish decision decreases the satisfaction of other users already associated with j^* and as a result, some of these users may wish to switch to some other lightly loaded AP for higher throughput/higher satisfaction. This naturally leads to the following game theoretic setting:

- **Players:** the users $i \in I$ are regarded as the players of the game.
- **Action space:** each user i has an action set J where action $j \in J$ corresponds to the decision of user i to associate with BS j . Therefore, the joint action set of all users is the Cartesian product $A = J \times \dots \times J$.
- **Cost functions:** the cost function of user i is the load $l_j + \frac{T_i}{C_{ij}}$ at the AP j after it decides to associate with AP j .

Since the cost function involves the term l_j dependent on the set of users taking action j , it follows that user costs are functions of the joint action of all the users.

Each user tries to selfishly minimize its cost by associating with the AP j^* that is least loaded upon its joining that AP as given by (4). Note from (4) that user i plays its best strategy taking other players' strategies as given. This is termed as user i performing an *improvement step* (thereby reducing its cost or equivalently increasing its satisfaction) by playing its *best response* and hence the name for the algorithm. Each user is activated one after the other to play its best response and this is continued until the system reaches a Nash equilibrium where there is no incentive for any user to unilaterally change its strategy. We first show that the best response algorithm indeed reaches a Nash equilibrium after a finite number of steps.

Proposition 1. *The best response algorithm converges to a Nash equilibrium after a finite number of improvement steps.*

Proof: Any assignment of users to APs induces a *sorted load vector* $(l_{[1]}, \dots, l_{[|K|]})$ where $l_{[j]}$ denotes the load on the machine that has the j -th highest load. If an assignment of users to APs is not a Nash equilibrium, then there exists a user which can perform an improvement step, i.e., reduce its cost by moving to another AP. We show that such an improvement step leads to a sorted load vector which is lexicographically smaller than the one preceding it. This proves that a Nash equilibrium is reached after a finite number of steps.

Suppose, given any sorted load vector $(l_{[1]}, \dots, l_{[|J|]})$, user i performs an improvement step and moves from AP j to AP k

where the indices are with respect to the positions of the APs in the sorted load vector. Clearly $k > j$. The improvement step decreases the load on j and increases the load on k . However, the increased load on k is still smaller than $l_{[j]}$, as, otherwise, user i would not decrease its cost. Thus, the number of APs with load at least $l_{[j]}$ is decreasing. Also, the load on all other APs with load at least $l_{[j]}$ remains unchanged. Consequently, the improvement step leads to a new sorted load vector which is lexicographically smaller than $(l_{[1]}, \dots, l_{[|J|]})$. ■

A centralized implementation of the best response algorithm could order the users in some arbitrary way and then activate one user at a time in that order to play its best response strategy. On the other hand, for a distributed implementation, each user could play its best response strategy with some probability p instead of playing it deterministically in order to avoid simultaneous user jumps leading to potential oscillations. A fully distributed, user-centric randomized algorithm along these lines is described and its convergence to a Nash equilibrium is proved in [3].

V. PERFORMANCE BOUNDS

In this section, we provide bounds on the performance of both the greedy algorithm and the best response algorithm. For the greedy algorithm, we provide an upper bound on the *competitive ratio* while for the best response algorithm, we provide an upper bound on the *price of anarchy*.

A. Competitive Analysis of the Greedy Algorithm

The greedy *online* algorithm makes decisions on-the-fly as new users arrive into the system. However, if the system were to wait long enough until all the users arrived, the optimal association could also be calculated *offline*. Competitive analysis is used to compare the performance of an online algorithm with the optimal offline solution. An online algorithm is *competitive* if its *competitive ratio* - the ratio between its performance and the optimal offline solution - is bounded.

Let OPT be the optimal makespan solved offline after all the users have arrived into the system and let l_j be the load at AP j in the online algorithm as defined in (3). Then, we define the competitive ratio c for the greedy online algorithm as follows:

$$c = \frac{\max_j l_j}{OPT}.$$

Let $l_{\max}(i)$ and $l_{\min}(i)$ denote the highest and the lowest load respectively among all APs after the i th user is assigned by the greedy algorithm. Let T_{\max} denote the maximum target rate among all users. Also, let C_{\min} denote the minimum peak rate and C_{\max} denote the maximum peak rate of an AP on the network. For example, consider a WiFi network where these peak rates correspond to the smallest and largest rates on the modulation and coding table of the 802.11 standard. The ratio $\beta = \frac{C_{\min}}{C_{\max}}$ is clearly bounded.

Lemma 1. *For a topology where every user i can connect to any AP j with a peak rate C_{ij} satisfying $C_{\min} \leq C_{ij} \leq C_{\max}$, after the i th user is assigned by the greedy algorithm, we have:*

$$l_{\max}(i) \leq l_{\min}(i) + \frac{T_{\max}}{C_{\min}} \quad (5)$$

for every i .

Proof: Firstly, we note that the inequality holds for $i = 1$. We then prove the induction step by showing that the inequality holds for i if it holds for $i - 1$. Suppose the inequality holds for $i - 1$, i.e., we have $l_{\max}(i - 1) \leq l_{\min}(i - 1) + \frac{T_{\max}}{C_{\min}}$. We now show that $l_{\max}(i) \leq l_{\min}(i) + \frac{T_{\max}}{C_{\min}}$ also holds after user i is assigned. We denote the AP of the lowest load before user i arrives as AP j . There are two cases after user i arrives: either user i will be assigned to AP j , or it will be assigned to one of the other APs.

1) *Case 1.1:* If user i is assigned to AP j , making AP j the new highest loaded AP (we denote its load as $l_j(i)$), then $l_j(i) \leq l_j(i - 1) + \frac{T_{\max}}{C_{\min}}$ since $T(i) \leq T_{\max}$ and $C_{ij} \geq C_{\min}$. At this moment, we have $l_{\max}(i) = l_j(i) \leq l_j(i - 1) + \frac{T_{\max}}{C_{\min}} \leq l_{\min}(i) + \frac{T_{\max}}{C_{\min}}$ since $l_{\min}(i) \geq l_{\min}(i - 1)$ (the new lowest load must be greater or equal to the old one since AP j is no longer the lowest loaded AP).

2) *Case 1.2:* If user i is assigned to AP j while AP j is not the new highest loaded AP, then clearly $l_{\max}(i) = l_{\max}(i - 1) \leq l_{\min}(i - 1) + \frac{T_{\max}}{C_{\min}} \leq l_{\min}(i) + \frac{T_{\max}}{C_{\min}}$ since again $l_{\min}(i) \geq l_{\min}(i - 1)$.

3) *Case 2.1:* Now consider the case where user i is not assigned to the lowest loaded AP j , but to some other AP k which becomes the new highest loaded AP. Inequality $l_k(i) \leq l_{\min}(i - 1) + \frac{T_{\max}}{C_{\min}}$ should hold because otherwise the greedy algorithm would have assigned user i to AP j instead of AP k . Also since in this case $l_{\min}(i) = l_{\min}(i - 1)$, clearly $l_{\max}(i) = l_k(i) \leq l_{\min}(i) + \frac{T_{\max}}{C_{\min}}$.

4) *Case 2.2:* At last, user i is assigned to some other AP k and AP k is not the new highest loaded AP. This means neither the highest loaded AP nor the lowest loaded AP has changed after user i arrived. Then clearly $l_{\max}(i) \leq l_{\min}(i) + \frac{T_{\max}}{C_{\min}}$ still holds this time. ■

We now derive an upper bound on the competitive ratio c through the following theorem:

Theorem 1. *The maximum load at any instant of the online algorithm satisfies*

$$\max_j l_j \leq \frac{1}{\beta} \left(1 + \mathcal{O} \left(\frac{|J|}{|I|} \right) \right) OPT \quad (6)$$

where $\beta = \frac{C_{\min}}{C_{\max}}$.

Proof: Consider the optimal offline solution and let l_j^* be the load at AP j in the optimal assignment. Since the maximum load OPT must be greater than the load averaged over all the APs in the optimal assignment, we have:

$$OPT \geq \frac{1}{|J|} \sum_{i=1}^{|J|} l_j^* = \frac{1}{|J|} \sum_{i=1}^{|I|} \frac{T_i}{C_{ij^*(i)}},$$

where $j^*(i)$ is the AP with which user i is associated in the greedy assignment. Since $\frac{T_i}{C_{ij^*(i)}} \geq \frac{T_i}{C_{\max}}$, we further have:

$$OPT \geq \frac{\sum_{i=1}^{|I|} T_i}{|J| C_{\max}}. \quad (7)$$

On the other hand, letting l_{\max} denote the maximum load in the greedy assignment and using Lemma 1, we have:

$$|J|l_{\max} \leq |J|l_{\min} + |J|\frac{T_{\max}}{C_{\min}}.$$

Since the minimum load l_{\min} is less than the load $\frac{1}{|J|}\sum_j l_j$ averaged across all the APs, we have:

$$\begin{aligned} |J|l_{\max} &\leq \sum_j l_j + |J|\frac{T_{\max}}{C_{\min}} \\ &= \sum_{i=1}^{|I|} \frac{T_i}{C_{ij^*(i)}} + |J|\frac{T_{\max}}{C_{\min}} \\ &\leq \sum_{i=1}^{|I|} \frac{T_i}{C_{\min}} + |J|\frac{T_{\max}}{C_{\min}}. \end{aligned} \quad (8)$$

We finally have

$$l_{\max} \leq \frac{\sum_{i=1}^{|J|} \frac{T_i}{C_{\min}} + |J|\frac{T_{\max}}{C_{\min}}}{|J|}. \quad (9)$$

Dividing (9) by (7), we obtain:

$$\frac{l_{\max}}{OPT} \leq \frac{C_{\max}}{C_{\min}} + \frac{C_{\max}}{C_{\min}} \frac{|J|}{\frac{\sum_{i=1}^{|I|} T_i}{T_{\max}}}.$$

Since $\frac{\sum_{i=1}^{|I|} T_i}{T_{\max}} = \mathcal{O}(|I|)$, we have:

$$\frac{l_{\max}}{OPT} \leq \frac{1}{\beta} \left(1 + \mathcal{O}\left(\frac{|J|}{|I|}\right) \right).$$

Corollary 1. *In the practically relevant regime of a highly loaded network where the number of users $|I|$ is very large compared to the number of APs $|J|$, the competitive ratio $c \leq \frac{1}{\beta}$.*

Proof: As the number of users is clearly much greater than the number of APs, the term $\mathcal{O}\left(\frac{|J|}{|I|}\right)$ in (6) is negligible yielding the constant $\frac{1}{\beta}$. ■

B. Upper Bound on the Price of Anarchy of the Best Response Algorithm

Price of anarchy is a popular measure used to study the inefficiency of Nash equilibria. It is defined as the ratio between the worst objective function value of a Nash equilibrium (among possibly multiple Nash equilibria) of a game and that of an optimal outcome [4]. In the user-AP association problem, the price of anarchy would be the worst case ratio between the makespan of a Nash equilibrium and the optimal makespan. In the following, we derive an upper bound on the price of anarchy of the best response algorithm.

We use the same notation as in Section V-A for ease of exposition although they denote different quantities in this subsection. Let OPT be the optimal makespan across the $|J|^{|K|}$ user-AP association configurations and let l_j denote the load of AP j at a Nash equilibrium attained by the best response algorithm. The price of anarchy p of the best response

algorithm is defined as the minimum makespan ratio across all Nash equilibria attained by the best response algorithm:

$$p = \min_{\text{Nash equilibria}} \frac{\max_j l_j}{OPT}.$$

At the Nash equilibrium, define L_k as the set of access points with load greater than or equal to $k \cdot OPT$ for every positive integer k , i.e.,

$$L_k = \{j : l_j \geq k \cdot OPT\} \quad (10)$$

We use $I(j)$ to denote the set of users associated with AP j and $I(L_k)$ be the set of users associated with any of the APs in L_k .

Lemma 2. *Given any AP j in L_{k+1} , for every user $i \in I(j)$ and for every AP $j' \in J \setminus L_k$ (i.e., all the APs in J but not in L_k),*

$$\frac{T_i}{C_{ij'}} > OPT.$$

Proof: Suppose on the contrary that there exists an AP j , a user $i \in I(j)$ which is assigned to AP j at the Nash equilibrium and an AP $j' \in J \setminus L_k$ such that $\frac{T_i}{C_{ij'}} \leq OPT$. Since $\frac{T_i}{C_{ij'}} \leq OPT$ by assumption, switching user i from AP j to AP j' would result in a load strictly less than $(k+1) \cdot OPT$. This is a contradiction because at a Nash equilibrium, i has no motivation to switch its association from j to j' . ■

Lemma 3. *Let $j(i)$ denote the access point to which user i is associated at the Nash equilibrium and let $j^*(i)$ denote the AP to which user i is associated in the optimal solution. Suppose i is a user such that $j(i) \in L_{k+1}$. Then $j^*(i) \in L_k$.*

Proof: Suppose on the contrary that $j^*(i) \in J \setminus L_k$. Then, from Lemma 2, the contribution $\frac{T_i}{C_{ij^*(i)}}$ of user i to the load at $j^*(i)$ exceeds OPT which contradicts the fact that in the optimal solution the maximum load among all APs is OPT . ■

Lemma 4. *When $\beta |c| \geq 10$ and $\beta \in [5 - \sqrt{24}, \frac{8}{11}]$, then given*

$$|J| \geq |c|! \beta^{|c|}$$

we'll have

$$|c| = \mathcal{O}\left(\frac{\log |J|}{\log \log |J|}\right).$$

Proof: See appendix. ■

Note that the somewhat awkward interval in which β has to lie in for the lemma to hold is the outcome of real analysis calculations required to bound the competitive ratio (see appendix). Also, note that β has to approximately be in $[0.1, 0.7]$ which is consistent with real-world peak rates per the 802.11 standard, see, for example, the data rates of the most recent incarnation of the standard, 802.11ac [18].

We now derive an upper bound on the price of anarchy p through the following theorem:

Theorem 2. *The maximum load among all the APs at any Nash equilibrium attained by the best response algorithm satisfies*

$$\max_j l_j = \max \left(\mathcal{O}\left(\frac{\log |J|}{\log \log |J|}\right), \frac{1}{\beta} \right) OPT \quad (11)$$

for

$$\beta \in [5 - \sqrt{24}, \frac{8}{11}]$$

Since (11) holds for any Nash equilibrium, the price of anarchy $p = \max \left(O \left(\frac{\log |J|}{\log \log |J|} \right), \frac{1}{\beta} \right)$.

Proof: Consider the set of access points L_{k+1} which have load greater than $(k+1) \cdot OPT$ at the Nash equilibrium.

For any $j \in L_{k+1}$, consider the set of users $I(j)$ assigned to AP_j at the Nash equilibrium. Since the total contribution of these users to the load of $AP_j \in L_{k+1}$ is at least $(k+1)OPT$, we have:

$$\sum_{i \in I(j)} \frac{T_i}{C_{ij}} \geq (k+1)OPT.$$

Using the fact that for every user i ,

$$\frac{T_i}{\min_{(i,j)} C_{ij}} \geq \frac{T_i}{C_{ij}} \quad \forall j$$

we further have:

$$\sum_{i \in I(j)} \frac{T_i}{\min_{(i,j)} C_{ij}} \geq (k+1)OPT$$

which implies

$$\sum_{j \in L_{k+1}} \sum_{i \in I(j)} T_i \geq |L_{k+1}| C_{\min} (k+1)OPT. \quad (12)$$

Now, from Lemma (3), every user i in $I(j)$ will be associated with an AP $j^* \in L_k$ in the optimal solution. Therefore,

$$\sum_{j \in L_{k+1}} \sum_{i \in I(j)} \frac{T_i}{C_{ij^*}} \leq |L_k| OPT. \quad (13)$$

Using the fact that for every user i

$$\frac{T_i}{C_{ij^*}} \leq \frac{T_i}{C_{\max}},$$

we have

$$\sum_{j \in L_{k+1}} \sum_{i \in I(j)} T_i \leq |L_k| C_{\max} OPT. \quad (14)$$

Thus, from (14) and (12), we have

$$|L_k| \geq \beta(k+1)|L_{k+1}|,$$

where $\beta = \frac{C_{\min}}{C_{\max}}$. Proceeding iteratively, we get

$$|L_0| \geq k! \beta^k |L_k|, \quad (15)$$

where $|L_0| = |J|$ is the number of APs in the network.

Let c be the ratio between the makespan at the Nash equilibrium and the makespan at the optimal solution, i.e.,

$$c = \frac{\max_j l_j}{OPT}$$

and let $\lfloor c \rfloor$ be the greatest integer less than or equal to c . From (15) and since $|L_{\lfloor c \rfloor}| \geq 1$, we have

$$|J| \geq \lfloor c \rfloor! \beta^{\lfloor c \rfloor}. \quad (16)$$

Now we use (16) to show that

$$\lfloor c \rfloor = \mathcal{O} \left(\frac{\log |J|}{\log \log |J|} \right). \quad (17)$$

We consider the following two cases.

1) *Case 1:* Firstly we consider the case where $\beta \lfloor c \rfloor \geq 10$. This has been proved in Lemma 4.

2) *Case 2:* Now we study the case where $1 < \beta \lfloor c \rfloor < 10$. Since function $f(\lfloor c \rfloor) = \lfloor c \rfloor! \beta^{\lfloor c \rfloor}$ is an increasing function when $\lfloor c \rfloor > \frac{1}{\beta}$, we denote $\lfloor c \rfloor^* = \left\lceil \frac{10}{\beta} \right\rceil$ and we have $f(\lfloor c \rfloor) < f(\lfloor c \rfloor^*), \forall \lfloor c \rfloor < \lfloor c \rfloor^*$. Since $f(\lfloor c \rfloor) \leq |J|$, there exists a positive real number S such that:

$$f(\lfloor c \rfloor) < f(\lfloor c \rfloor)S = f(\lfloor c \rfloor^*) \leq |J|S,$$

where $S \leq \frac{\left\lceil \frac{10}{\beta} \right\rceil! \beta^{\left\lceil \frac{10}{\beta} \right\rceil}}{\left\lfloor \frac{1}{\beta} \right\rfloor! \beta^{\left\lfloor \frac{1}{\beta} \right\rfloor}}$. Thus, we have

$$\lfloor c \rfloor < \lfloor c \rfloor^* \leq \mathcal{O} \left(\frac{\log |J| S}{\log \log |J| S} \right).$$

Since S is a constant, it follows that

$$\lfloor c \rfloor < \mathcal{O} \left(\frac{\log |J|}{\log \log |J|} \right).$$

Therefore, (17) also holds for $\lfloor c \rfloor < \frac{10}{\beta}$. ■

VI. SIMULATION RESULTS

In this section, we evaluate the performance of the greedy and best response algorithms. We run two sets of simulations. In the first, we focus on real-world scenarios and compare the performance of our algorithms against the standard 802.11 scheme and simple extensions of it. In the second, we compare our schemes against the performance of the offline optimal scheme for a few characteristic scenarios.

A. Methodology

We conduct our simulations in MATLAB. We create a canonical topology resembling a large conference hall / convention center of size 120x100m. We deploy a variable number of APs uniformly inside the conference hall and assign to each one of them one out of four non-overlapping channels, illustrated by 4 different colors in Fig. 1. We choose to assume the existence of 4 non-overlapping channels based on the available spectrum and the typical channel widths used in 802.11 today. Users join the network at Poisson times and at random locations having different target rates, and stay at the network for an exponential amount of time. Upon the arrival of a new user, the user is assigned to an AP following the various user-AP association schemes under study. In the case of the greedy algorithm, a user will stay in the same AP throughout his/her stay at the network, whereas in the case of the best response algorithm, whenever a new user arrives or an existing user departs the network, users may switch from one AP to another until the system reaches a Nash equilibrium. Last, the SINRs between APs and users are computed based on path loss which solely depends on distance.

We introduce a ‘‘concentration factor’’ to vary the distribution of the users’ location from uniform all the way to situations where the vast majority of users gather in the center of the conference hall. Fig. 1 illustrates four topologies with different concentration levels and number of APs.

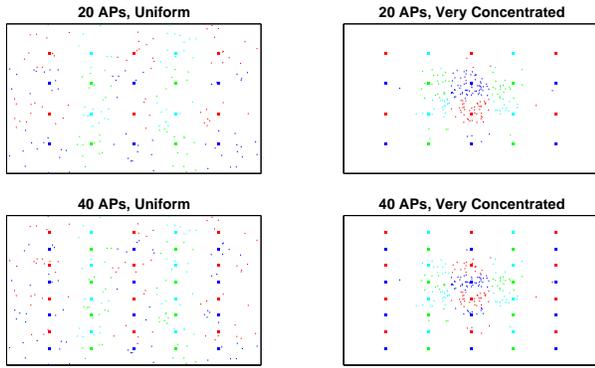


Fig. 1: Topologies with different concentration factors and number of APs.

B. Improvement Over Current Mechanisms

We measure the average data rate and the minimum satisfaction ratio of the users under difference scenarios and schemes.

We consider four schemes: the greedy scheme, the best response scheme, the original 802.11 scheme where users connect to the AP with the highest RSSI, and a simple extension of the original 802.11 scheme used by some manufacturers today where heavy loaded APs may reject a request from a user to connect to them if they serve a lot of users already, forcing the user to select the AP with the second or third highest RSSI value. We refer to this scheme as “802.11 with max users”. The topology used in this simulation is a conference hall with 20 APs uniformly deployed, and 100 initial users. New users join the network while old users leave, both in a Poisson process manner. We ran the simulation for 50 time units with different user arrival rates and concentration levels.

Impact on rates. We record for each user his/her rate during the period of time he/she is in the network, and compute the average rate over all users. We compare the greedy algorithm against the original 802.11 scheme for a variable number of APs and a number of concentration factor values. Note that both of these schemes do not switch users between APs once a user is associated with a specific AP.

Fig. 2 shows that the greedy algorithm outperforms the original scheme by a sizable amount, especially when the user concentration is high and the number of users is not too large. Both of these are to some extent expected, as high levels of user concentrations in the center of a room create large imbalances between the load of different APs and the original 802.11 scheme is rather inefficient, and, large number of users push the network to reach its operational capacity in which case there is little room for rate improvements.

Impact on fairness. We record for each user his/her satisfaction ratio at each time slot, and report the minimum satisfaction ratio among all users.

We compare the greedy and best response schemes against the original 802.11 and the 802.11 with max users schemes. As shown in Fig. 3, the greedy and best response schemes achieve a better minimum satisfaction ratio than both the

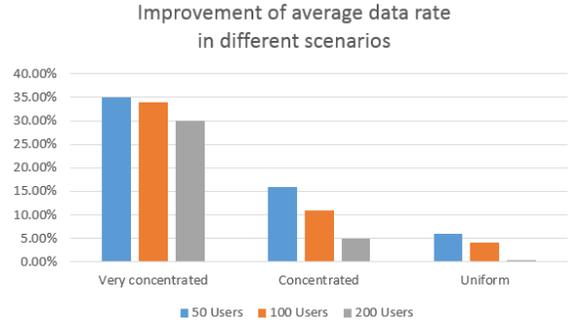


Fig. 2: Percentage of data rate increase using greedy algorithm over original 802.11.

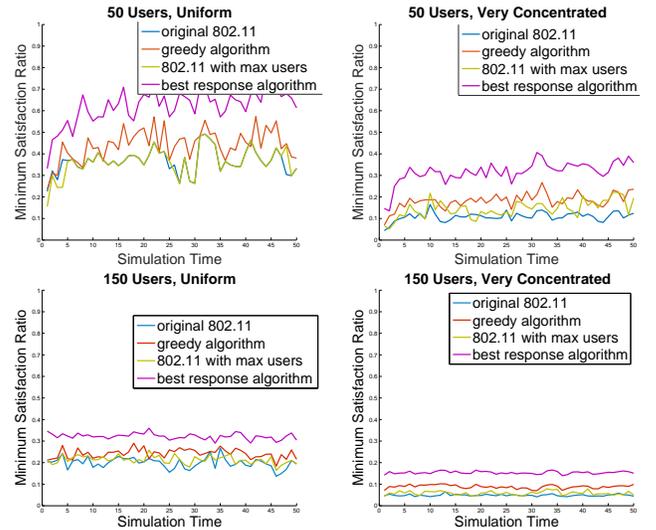


Fig. 3: Minimum satisfaction ratio in different topologies.

original 802.11 and the 802.11 with max users schemes for a wide range of scenarios. This is to be expected since our two schemes have been designed with this goal in mind. Note that in practice, it is quite important to take into account users’ target rates and satisfaction ratios because when a user receives significantly lower rate than his/her target rate he/she will most likely leave the network. For example, consider a user who wishes to watch a video yet receives service which cannot sustain the video playback. Thus, an improvement of the satisfaction ratios of users can have a sizable impact in the quality of experience of users.

C. Evaluation of optimality

We compare our schemes to the optimal one. Finding the optimal association requires solving an NP-hard integer optimization problem. Thus, we relax the problem by working with real numbers as follows:

$$\min_{x_{ij}} \max\{\hat{l}_1, \hat{l}_2, \hat{l}_3, \dots, \hat{l}_{|J|}\}$$

s.t.

$$\sum_{j=1}^{|J|} x_{ij} = 1 \quad \forall i$$

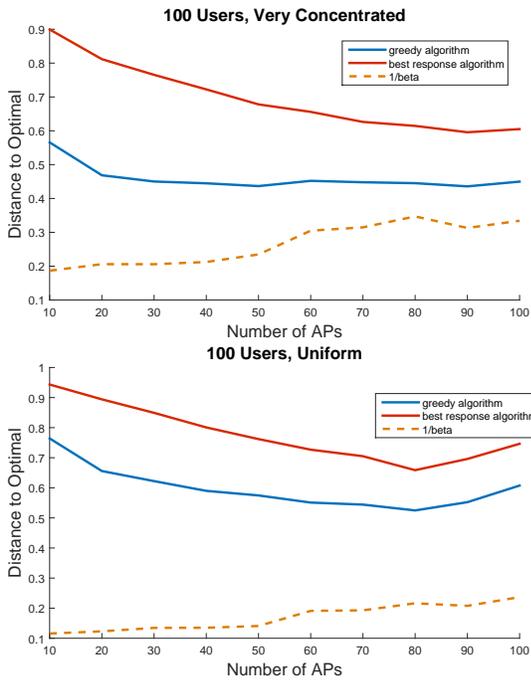


Fig. 4: Distance to optimal vs. number of APs.

$$\hat{l}_j = \sum_{i=1}^{|I|} \frac{T_i}{C_{ij}} x_{ij}$$

$$x_{ij} \in [0, 1] \quad \forall i, j$$

where \hat{l}_j is the load at AP j when fractional associations are allowed and $x_{ij} \in [0, 1]$ represents the fraction of time user i is served by AP j . $|I|$ represents the total number of users, and $|J|$ represents the total number of APs.

We run the simulation for 50 time units, with an arrival rate set to 100 and a user concentration level set to a high value. We vary the number of APs in the system from 10 to 100. We record the minimum satisfaction ratio achieved by our algorithms and divide it by the minimum satisfaction ratio achieved by the relaxed optimal association. We refer to this ratio as the “distance to optimal” and plot its value in Fig. 4. As shown in the figure, the best response algorithm performs within 5 to 20% off the optimal whereas the greedy one performs within 20 to 50%. Recall that the optimal scheme is offline, thus it is expected that any online scheme will have a sizable performance gap against it. Also, the actual performance gap is, of course, less than the theoretical bound $\frac{1}{\beta}$ derived in Section V.

D. Importance of Target Rates

In this section we study the implication of target rates. Without loss of generality, we consider two types of users, users who check their emails and users who stream video online. Users of the same type are assumed to have the same target rate, and, of course, the users in the latter category have a higher target rate than the users in the former one. We vary the ratio of the target rates from 10:20 to 10:400 while maintaining the total load of the system at about 150%. Similarly to before,

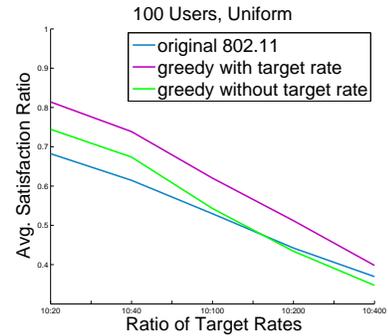


Fig. 5: Average satisfaction ratio vs. ratio of target rates.

we consider a scenario with 20 APs, a user arrival rate equal to 100, and a user concentration level set to uniform.

We measure the average satisfaction ratio over all users achieved by three different algorithms: the original 802.11 scheme, our greedy algorithm which takes into account the user target rates, and, a greedy algorithm which is agnostic to the target rates. Specifically, according to the greedy algorithm without target rates, a user i which associates with an AP j with a total of associated users N_j , gets $1/N_j$ of the airtime of the AP j and a rate equal to C_{ij}/N_j . That is, $\alpha_{ij} = 1/n$ instead of the value in Eq. (1).

As shown in Fig. 5, our greedy algorithm outperforms the other two since they do not consider the users’ target rates. Interestingly, a greedy algorithm which is agnostic to target rates performs worse than the original 802.11 scheme as the variance of users’ target rates grows. As already mentioned, the user satisfaction is an important measure in practice, since some applications may not even operate in the absence of a minimum rate.

VII. PRACTICAL CONSIDERATIONS

In this section we discuss how to implement the suggested algorithms in a real-world setup. For ease of exposition, we first assume that minor changes to the association procedures of the 802.11 protocol stack are allowed both at the client and the AP side. Then, we describe how to implement the schemes under the more realistic scenario where only APs software/firmware can be altered, whereas no changes are feasible at the client side.

Neighboring APs. In today’s enterprise WiFi networks, a client associates with the AP from which it receives the highest RSSI reading. (RSSI is a real-world mechanism to estimate the received signal strength and thus the SINR.) If this AP rejects the association request, for example, because the AP in question has reached the maximum allowed number of clients, then the client requests to be associated with the AP from which it receives the second highest RSSI reading. In general, there is a finite set of APs from which a client is receiving an acceptably high RSSI reading. We call an AP a neighboring AP of client i if the RSSI reading from this AP is above a threshold. (For example, this threshold may simply be the minimum received power that would allow to select the lowest rate option from the 802.11 modulation and coding table.)

Association Decision. We require that all APs broadcast to their potential clients their load information. A client i will record for each neighboring AP j the RSSI $_{ij}$ value, and will receive the AP load value l_j . Based on this information, a client calculates the potential target load value l_j^{new} of any neighboring AP if the new client were to be assigned to it using Eq. (3). (The client is obviously aware of its target rate T_i). Finally, the client chooses the AP with the lowest l_j^{new} as described in Eq. (4) and sends an association request to it. Per the greedy algorithm, every new client will perform the above procedure once. Also, if an AP de-associates a client for whatever reason, the client would perform the above procedure to find a new AP to associate with.

User Switching. In the case of the best response algorithm a client may disconnect from one AP and connect to another one if conditions change enough such that the new AP would have a lower load after the client connects to it, than the load of the AP to which the client is currently connected. As clients switch, AP loads change and other clients may choose to switch till a Nash equilibrium is reached. Due to the overhead from such association/de-association steps and as discussed already, the best response scheme is of algorithmic interest rather than of real-world, practical interest.

Backward compatibility. It is not likely that clients will change their 802.11 stack to accommodate new association schemes unless such changes become part of a new standard. It is however feasible for infrastructure nodes to do so, since all APs of an enterprise-WiFi network are controlled and managed by the same entity. That said, any such change should be backward compatible and the clients should be agnostic to it. The suggested greedy algorithm can be implemented in such a backward compatible manner by taking advantage of the existing feature of the 802.11 protocol which allows APs to reject an association request from a client. Specifically, APs can record the RSSI values between themselves and all their clients and estimate the target rates of their clients. Then, APs may exchange with nearby APs the RSSI and target rate values as well as their own load. With this information, each AP can compute the AP that satisfies Equation (4), and reject an association request from a client if it is not the “right” AP. This would force the client to send an association request to another AP, and so on till the “right” AP receives the association request which it will accept.

APPENDIX A

In order to prove Lemma 4, we first assume $\beta \lfloor c \rfloor \geq 10$ and show that $\lfloor c \rfloor! \beta^{\lfloor c \rfloor} \geq \lfloor \beta \lfloor c \rfloor \rfloor!$.

$$\begin{aligned} \lfloor c \rfloor! \beta^{\lfloor c \rfloor} &= (\beta \lfloor c \rfloor) (\beta (\lfloor c \rfloor - 1)) (\beta (\lfloor c \rfloor - 2)) \\ &\quad \dots (\beta \lfloor \frac{1}{\beta} \rfloor) (\beta (\lfloor \frac{1}{\beta} \rfloor - 1)) \dots (\beta \cdot 1) \end{aligned} \quad (18)$$

which has $\lfloor c \rfloor$ terms in total and can be rewritten as:

$$\begin{aligned} \lfloor c \rfloor! \beta^{\lfloor c \rfloor} &= \boxed{(\beta \lfloor c \rfloor)} (\beta (\lfloor c \rfloor - 1)) (\beta (\lfloor c \rfloor - 2)) \dots \boxed{(\beta (\lfloor c \rfloor - \lfloor \frac{1}{\beta} \rfloor))} \\ &\quad (\beta (\lfloor c \rfloor - \lfloor \frac{1}{\beta} \rfloor - 1)) \dots \boxed{(\beta (\lfloor c \rfloor - 2 \lfloor \frac{1}{\beta} \rfloor))} \dots \\ &\quad \boxed{(\beta (\lfloor \frac{1}{\beta} \rfloor + 1))} (\beta (\lfloor \frac{1}{\beta} \rfloor)) (\beta (\lfloor \frac{1}{\beta} \rfloor - 1)) \dots (\beta \cdot 1). \end{aligned} \quad (19)$$

The terms from (19) shown in boxes are greater or equal to each of the terms in $\lfloor \beta \lfloor c \rfloor \rfloor!$:

$$\beta \lfloor c \rfloor \geq \lfloor \beta \lfloor c \rfloor \rfloor$$

$$\beta (\lfloor c \rfloor - \lfloor \frac{1}{\beta} \rfloor) = \beta \lfloor c \rfloor - \beta \lfloor \frac{1}{\beta} \rfloor \geq \lfloor \beta \lfloor c \rfloor \rfloor - 1$$

$$\beta (\lfloor c \rfloor - 2 \lfloor \frac{1}{\beta} \rfloor) = \beta \lfloor c \rfloor - 2\beta \lfloor \frac{1}{\beta} \rfloor \geq \lfloor \beta \lfloor c \rfloor \rfloor - 2$$

⋮

$$\beta (\lfloor \frac{1}{\beta} \rfloor + 1) = \beta \lfloor \frac{1}{\beta} \rfloor + \beta \geq 1$$

Note that we have a total of $\lfloor \beta \lfloor c \rfloor \rfloor$ such terms in (19).

Now we need to show the product of the remaining $\lfloor c \rfloor - \lfloor \beta \lfloor c \rfloor \rfloor$ terms in (19) is greater or equal to 1 so that the inequality $\lfloor c \rfloor! \beta^{\lfloor c \rfloor} \geq \lfloor \beta \lfloor c \rfloor \rfloor!$ holds. Clearly there are at most $\lfloor \frac{1}{\beta} \rfloor$ terms which are less than or equal to 1 in (19), which are $\beta, 2\beta, \dots, (\beta \lfloor \frac{1}{\beta} \rfloor - 1), \beta \lfloor \frac{1}{\beta} \rfloor$. Therefore if we can find another $\lfloor \frac{1}{\beta} \rfloor$ terms which are greater than 1 in the remaining part, and construct the following $\lfloor \frac{1}{\beta} \rfloor$ pairs with those terms which are less than or equal to 1 such that each pairwise product is greater or equal to 1, then clearly the product of all the remaining $(\lfloor c \rfloor - \lfloor \beta \lfloor c \rfloor \rfloor)$ terms would be proved to be greater or equal to 1 as well. The $\lfloor \frac{1}{\beta} \rfloor$ terms greater than 1 that we pick are $\beta (\lfloor c \rfloor - 1), \beta (\lfloor c \rfloor - 2), \dots, \beta (\lfloor c \rfloor - \lfloor \frac{1}{\beta} \rfloor + 1)$ and $\beta (\lfloor c \rfloor - \lfloor \frac{1}{\beta} \rfloor - 1)$. We thus construct the pairwise products as:

$$\beta (\lfloor c \rfloor - 1) \beta$$

$$\beta (\lfloor c \rfloor - 2) 2\beta$$

⋮

$$\beta (\lfloor c \rfloor - \lfloor \frac{1}{\beta} \rfloor + 1) \beta (\lfloor \frac{1}{\beta} \rfloor - 1)$$

$$\beta([c] - \left\lfloor \frac{1}{\beta} \right\rfloor - 1)\beta\left(\left\lfloor \frac{1}{\beta} \right\rfloor\right)$$

First, we study the first pairwise product. The following inequality holds since $\beta[c] \geq 10$:

$$\beta([c] - 1)\beta = \beta^2[c] - \beta^2 \geq 10\beta - \beta^2. \quad (20)$$

Also, the inequality

$$10\beta - \beta^2 \geq 1 \quad (21)$$

holds when $\beta \in [5 - \sqrt{24}, 5 + \sqrt{24}]$. Thus when $\beta \in [5 - \sqrt{24}, 5 + \sqrt{24}]$, by (20) we have $\beta([c] - 1)\beta \geq 1$.

Now, we take a look at the remaining pairwise products. In general, for any integer $i \in [2, \left\lfloor \frac{1}{\beta} \right\rfloor - 1]$, the inequality

$$\beta([c] - i)i\beta = i\beta^2[c] - i^2\beta^2 \geq 10i\beta - i^2\beta^2$$

holds because again $\beta[c] \geq 10$. And since we already know that (21) holds when $\beta \in [5 - \sqrt{24}, 5 + \sqrt{24}]$, we want to solve for the conditions under which the inequality

$$10i\beta - i^2\beta^2 \geq 10\beta - \beta^2 \quad (22)$$

holds so that $\beta([c] - i)i\beta \geq 10i\beta - i^2\beta^2 \geq 10\beta - \beta^2 \geq 1$. The (22) can be rewritten as:

$$10\beta(i - 1) \geq (i^2 - 1)\beta^2$$

which holds when:

$$0 \leq \beta \leq \frac{10}{i + 1}.$$

For $\beta \geq 5 - \sqrt{24} > 0.1$, we have $i \leq \left\lfloor \frac{1}{\beta} \right\rfloor - 1 \leq 9$, which leads to $\frac{10}{i+1} \geq 1$. This implies that (22) holds when $5 - \sqrt{24} \leq \beta \leq 1$. As (21) also holds when $5 - \sqrt{24} \leq \beta \leq 1$, the inequality

$$\beta([c] - i)i\beta \geq 1$$

holds when $5 - \sqrt{24} \leq \beta \leq 1$.

Finally, for the last pairwise product we have:

$$\begin{aligned} & \beta([c] - \left\lfloor \frac{1}{\beta} \right\rfloor - 1)\beta\left(\left\lfloor \frac{1}{\beta} \right\rfloor\right) \\ &= (\beta^2[c] - \beta^2\left\lfloor \frac{1}{\beta} \right\rfloor - \beta^2)\left\lfloor \frac{1}{\beta} \right\rfloor \\ &\geq (10\beta - \beta^2\left\lfloor \frac{1}{\beta} \right\rfloor - \beta^2)\left\lfloor \frac{1}{\beta} \right\rfloor \\ &= 10\beta\left\lfloor \frac{1}{\beta} \right\rfloor - \beta^2\left\lfloor \frac{1}{\beta} \right\rfloor^2 - \beta^2\left\lfloor \frac{1}{\beta} \right\rfloor \\ &\geq 10\beta\left(\frac{1}{\beta} - 1\right) - 1 - \beta \\ &= 9 - 11\beta. \end{aligned} \quad (23)$$

Since the inequality $9 - 11\beta \geq 1$ holds when $\beta \leq \frac{8}{11}$, the last pairwise product is also greater or equal to 1 when $\beta \in [0, \frac{8}{11}]$.

Therefore, when $\beta \in [5 - \sqrt{24}, \frac{8}{11}]$ and $\beta[c] \geq 10$ we have:

$$\begin{aligned} & |J| \geq [c]!\beta^{[c]} \geq [\beta[c]]! = \Gamma([\beta[c]] + 1) \geq \Gamma(\beta[c]) \\ &\implies \beta[c] \leq \Gamma^{-1}(|J|) \\ &\implies \beta[c] \leq \Theta(\log |J| / \log \log |J|) \\ &\implies [c] \leq \frac{1}{\beta} \Theta(\log |J| / \log \log |J|). \end{aligned} \quad (24)$$

REFERENCES

- [1] Cisco, "Cisco Wireless LAN Controller Configuration Guide," Cisco Systems, Inc., Tech. Rep., 2013.
- [2] J. G. Andrews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, "An overview of load balancing in HetNets: Old myths and open problems," *arXiv preprint arXiv:1307.7779*, 2013.
- [3] D. Bethanabhotla, O. Y. Bursalioglu, H. C. Papadopoulos, and G. Caire, "Optimal user-cell association for massive MIMO wireless networks," *CoRR*, vol. abs/1407.6731, 2014. [Online]. Available: <http://arxiv.org/abs/1407.6731>
- [4] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge University, 2007.
- [5] H. Ishii, Y. Kishiyama, and H. Takahashi, "A novel architecture for LTE-B: C-plane/U-plane split and phantom cell concept," in *IEEE Globecom Workshops*, 2012, pp. 624–630.
- [6] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. on Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, 2013.
- [7] P. Bahl, M. Hajiaghayi, K. Jain, S. Mirrokni, L. Qiu, and A. Saberi, "Cell breathing in wireless LANs: Algorithms and evaluation," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 2, pp. 164–178, Feb 2007.
- [8] Y. Bejerano and S. jae Han, "Cell breathing techniques for load balancing in wireless LANs," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 6, pp. 735–749, June 2009.
- [9] G. Athanasiou, P. C. Weeraddana, C. Fischione, and L. Tassiulas, "Optimizing client association in 60 GHz wireless access networks," *arXiv preprint arXiv:1301.2723*, 2013.
- [10] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless LANs using association control," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. ACM, 2004, pp. 315–329.
- [11] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *Proc. IEEE INFOCOM*, 2008, pp. 1004–1012.
- [12] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, "RAT selection games in HetNets," in *Proc. IEEE INFOCOM*, April 2013, pp. 998–1006.
- [13] A. Goel, A. Meyerson, and S. Plotkin, "Approximate majorization and fair online load balancing," *ACM Trans. Algorithms*, vol. 1, no. 2, pp. 338–349, Oct. 2005. [Online]. Available: <http://doi.acm.org/10.1145/1103963.1103970>
- [14] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, "On-line routing of virtual circuits with applications to load balancing and machine scheduling," *J. ACM*, vol. 44, no. 3, pp. 486–504, May 1997. [Online]. Available: <http://doi.acm.org/10.1145/258128.258201>
- [15] J. K. Lenstra, D. B. Shmoys, and É. Tardos, "Approximation algorithms for scheduling unrelated parallel machines," *Mathematical programming*, vol. 46, no. 1-3, pp. 259–271, 1990.
- [16] N. Andelman, M. Feldman, and Y. Mansour, "Strong price of anarchy," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 189–198.
- [17] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman, "Beacon-stuffing: Wi-Fi without associations," in *IEEE Workshop on Mobile Computing Systems and Applications, HotMobile*, 2007, pp. 53–57.
- [18] "IEEE 802.11ac - Wikipedia," https://en.wikipedia.org/wiki/IEEE_802.11ac, Last Updated: January 14, 2014.