2017 29th International Teletraffic Congress

Load-driven cell assignment algorithms for dense pico-cell networks

Bart Post
dept. of Electrical Engineering
Eindhoven University of Technology
Eindhoven, The Netherlands
Email: b.post@tue.nl

Sem Borst
dept. of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
Email: s.c.borst@tue.nl

***Abstract*—Fuelled by the proliferation of smartphones, wireless traffic has experienced huge growth, which will continue with the emergence of ultra-broadband 5G applications, and exacerbate the capacity strain in cellular networks. Deployment of pico access points, reducing cell sizes and allowing more efficient reuse of limited radio spectrum, provides a powerful approach to cope with traffic hot spots and bring capacity relief. This network densification makes cell planning more challenging though, and tends to result in more irregular cells with possibly overlapping coverage areas and greater variability in traffic loads. This raises a critical need for more intelligent cell selection algorithms, which not only take signal strength values into account, but also load conditions in order to harness the full potential of the pico-cells.**

In the present paper we analyse online cell selection algorithms that use a parsimonious set of load-driven control parameters to determine an optimal user association in a measurement-based manner, without requiring explicit knowledge of the system parameters. We exploit stochastic approximation techniques to establish the convergence of the control parameters to the optimal values. Extensive simulation experiments for scenarios with many pico access points confirm that the algorithms are quite effective in optimally balancing the traffic loads in hot spot areas, and further demonstrate that they substantially outperform conventional approaches in terms of service denials and low throughput percentiles. We consider several implementation options and evaluate the relative benefits and potential trade-offs.

***Keywords*-wireless pico-cell networks; load balancing; optimal cell selection; stochastic approximation;**

I. Introduction

Wireless cellular networks have experienced immense growth in traffic loads over the last few years fuelled by the rapid proliferation of smartphones and bandwidth-hungry applications. The sharp rise in traffic volumes is widely forecast to continue and exacerbate the capacity strain in cellular networks.

Two key options to support further growth within the confinements of the available radio spectrum are to implement multi-antenna techniques and/or deploy pico access points so as to cover traffic hot spots. Both options provide particularly powerful approaches in conjunction with optical backhaul links, enabling radio-over-fibre technologies [1], [2].

In the present paper we focus on the deployment of pico access points, which yields a significant potential for capacity gains by reducing cell sizes and allowing higher spectral reuse and efficiency. The denser concentration of access points also raises challenging issues though, in particular with regard to cell planning and traffic engineering [3], [4].

For example, physical constraints will typically make it even harder to arrange pico access points in an ideal hexagonal pattern, which causes the coverage areas to significantly overlap, and the natural cell regions to be irregularly shaped. Due to the smaller and less regular cell regions, the nominal traffic loads will tend to exhibit not only more spatial variation but also stronger temporal fluctuations (burstiness). This variability in traffic could potentially result in severe load imbalances and performance degradation in case users are simply assigned to the pico access point that offers the best signal strength. On a positive note, the higher degree of overlapping coverage areas also offers greater flexibility in a assigning users to access points.

In conclusion, pico cell deployments create both a stronger need and greater scope for more intelligent user association algorithms which not only take signal strength values into account but also load conditions. The design of such load-aware cell selection algorithms is critical for effective pico cell deployments, capitalizing on the full capacity gains and ensuring excellent user-perceived performance.

In order to design load-aware cell selection algorithms, we formulate the problem of optimally balancing traffic loads as a linear program (LP). Since the LP formulation involves several system parameters that tend to be time-varying and hard to estimate, we analyse an online cell selection algorithm that solves the LP and determines the optimal user association in a measurement-based manner without requiring explicit knowledge of the system parameters.

This paper extends [5] in two ways. First, we will exploit stochastic approximation techniques to establish convergence of the proposed algorithm under suitable assumptions. Second, we present extensive simulation experiments for large systems. The simulation results also provide empirical evidence of the convergence, and confirm that the proposed algorithm is quite effective in optimally balancing the traffic loads. Comparisons further demonstrate that it substantially outperforms conventional approaches in terms of user-perceived throughput performance and service denials.

978-0-9883-0453-6/17 $31.00 © 2017 ITC
DOI 10.1109/ITC.2017.10

A. Discussion and related work

From a high-level perspective, the optimal cell selection may be viewed as a load balancing problem in a parallel-server system, where incoming jobs are assigned to one of several servers so as to distribute the load and optimize some performance metric of interest. The above-described problem formulation involves however a key feature which provides a fundamental distinction with the typical load balancing framework in the literature.

In particular, in conventional load balancing settings, servers may have different processing rates and there may be heterogeneous user classes with different service requirement distributions, but the service rate is determined by the server only, and not by the user identity. In our set-up, the service rate can depend on the specific combination of the user class and the access point in any arbitrary way. This is a manifestation of the different signal strengths of various users with respect to different access points, and creates a further challenge for load balancing policies [6].

The papers [7] and [8] also consider load balancing scenarios where the service rates depend on both server and users. However their proposed algorithms require solving an optimization problem for given rate requirements at each arrival or departure epoch. We on the other hand, focus on transfers with elastic rates and avoid solving the optimization problem at each arrival or departure epoch.

We note that several papers have addressed the problem of optimal user association from a utility maximization perspective [9]–[16]. However, these papers focus developing efficient approximation algorithms for static user ensembles, while we focus on optimizing the perceived throughput performance for dynamic populations of flows.

The optimal cell selection problem is also tangentially related to assignment schemes in dynamic spectrum access [17] and so-called congestion games, where the user population is however either supposed to be entirely static, or only subject to exogenous random variation [18]. In our situation, the variation in the population of elastic transfers is intrinsic and strongly impacted by the cell selection decisions. Ignoring the latter knock-on effects can yield highly suboptimal decisions and carry severe performance penalties [19].

B. Organization of the paper

The remainder of the paper is organized as follows. In Section II we present a detailed model description and introduce some useful notation and terminology. In Section III we state the optimization objective that we pursue and derive optimality results that will play a crucial role in the design of an online solution algorithm. In Section IV we specify the online solution algorithm, provide an interpretation, and discuss some crucial implementation aspects. In Section V we use stochastic approximation techniques to prove that the iterates of the control parameters of the algorithm converge to optimal values. In Section VI we provide extended simulation results that corroborate the convergence and show that our algorithm outperforms conventional methods. Finally, in Section VII we make some concluding remarks and provide suggestions for further research.

II. Model description

In this section we present a detailed model description and introduce some useful notation and terminology. We consider the downlink of wireless network with L pico access points (APs), and we focus on a scenario with elastic traffic. Some APs may be physically co-located, but involve different radio access technologies (e.g. 4G LTE or 5G with multi-mode handsets). Users initiate file transfers as a Poisson process of rate λ. When a user initiates a file transfer, it must be immediately and irrevocably assigned to one of the APs. For convenience, we assume that there is a discrete set of N user locations, which may be interpreted as a suitable discretization of the overall coverage area. A location does not necessarily have a geographical interpretation, but rather represents a class of users that all have (approximately) equal physical transmission rate characteristics with respect to the APs as described below. Denote by p_n the fraction of users with location n, with $\sum_{n=1}^{N} p_n = 1$, and by $\lambda_n = \lambda p_n$ the arrival rate of users in that location. The sizes of the file transfers initiated by users in location n are independent and have mean β_n.

Let $R_{n,l}$ be the physical transmission rate in bits per time unit received by a user at location n from AP l. By physical rate we mean the achievable transmission rate per resource unit (e.g. physical resource blocks in LTE, or frame durations in WiFi), multiplied with the total number of available resource units per unit of time. For a given user location n, the transmission rate $R_{n,l}$ may be zero for some (and in fact many) APs l, reflecting that location n falls outside the maximum transmission range of these APs. Thus, in typical scenarios, only a few of the $R_{n,l}$ values for a given location n will be non-zero, and these values can be reasonably well estimated from Signal-To-Interference-plus-Noise Ratio (SINR) measurements which are already common in current systems at the initiation of a flow.

Remark 1: For convenience, we assume that the physical transmission rates are constant over time. Time-varying rates due to fading and opportunistic scheduling gains are thus not explicitly considered here. We also implicitly assume that the physical transmission rates are not strongly impacted by time-varying activity levels at neighbouring access points.

Remark 2: We do not explicitly account for hand-offs among APs, which is reasonable when the time scale of the file transfers is relatively short compared to the mobility dynamics of the users, as would normally be the case in pico-cell deployments.

III. Optimization framework

In this section we state the optimization objective that we pursue in the form of a Linear Program and present the corresponding Lagrangian dual problem, which will play a crucial role in the design of an online solution algorithm.

Let $x_{n,l}$ be the long-term fraction of users at location n which are assigned to AP l, with $\sum_{l=1}^{L} x_{n,l} = 1$, $\forall n = 1, \ldots, N$. Define the long-term load of AP l as

$$\rho_l = \rho_l(\mathbf{x}) = \sum_{n=1}^{N} \frac{\lambda_n \beta_n}{R_{n,l}} x_{n,l}, \tag{1}$$

where $\mathbf{x} = (x_{n,l})_{n=1,\ldots,N,l=1,\ldots,L}$ is the vector of assignment fractions. For notational brevity, we write $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_L)$ for the vector of loads. A proper assignment vector $\mathbf{x}$ and its corresponding load vector $\boldsymbol{\rho}(\mathbf{x})$ are said to be Pareto-optimal if there exists no other proper assignment vector $\mathbf{x}'$ with $\boldsymbol{\rho}(\mathbf{x}') \leq \boldsymbol{\rho}(\mathbf{x})$, and strict inequality $\rho_l(\mathbf{x}') < \rho_l(\mathbf{x})$ for at least one AP l.

In this paper we focus on minimizing the maximum weighted load $\max_l\{w_l \rho_l\}$ with weights w_l for the various APs. This can be viewed as a weighted load-balancing problem and can be formulated by the Linear Program (LP) (2)-(5) [5], where the assignment fractions $x_{n,l}$ are the optimization variables.

$$\min_{\mathbf{x}} \quad U \tag{2}$$

$$\text{sub:} \quad U \geq w_l \rho_l = w_l \sum_{n=1}^{N} \frac{\lambda_n \beta_n}{R_{n,l}} x_{n,l}, \quad \forall l, \tag{3}$$

$$\sum_{l=1}^{L} x_{n,l} = 1, \qquad \forall n, \tag{4}$$

$$x_{n,l} \geq 0, \qquad \forall n, \forall l. \tag{5}$$

While the above-stated LP has a relatively simple form, it cannot be solved in a direct manner since in practice the values of λ_n and β_n are typically unknown and hard to estimate. In the next section we will therefore analyse an algorithm for solving the above-stated problem in an online fashion using load measurements and knowledge of the $R_{n,l}$ values only. As mentioned earlier, the $R_{n,l}$ values can be reasonably well estimated from SINR measurements.

The online algorithm relies on some structural properties of the optimal assignment fractions, which we will now describe. The optimal dual variables corresponding to the constraints defined by (3) are a solution of the Lagrangian Dual Problem (LDP) given by:

$$\max_{\mathbf{y}} \quad V^*(\mathbf{y}), \tag{6}$$

$$\text{sub:} \quad \sum_{l=1}^{L} y_l = 1, \quad \forall l, \tag{7}$$

$$y_l \geq 0, \qquad \forall l, \tag{8}$$

where $V^*(\mathbf{y})$ is the optimal value of the following sub-problem:

$$\min_{\mathbf{x}} \quad V(\mathbf{y}) = \sum_{l=1}^{L} y_l w_l \rho_l(\mathbf{x}), \tag{9}$$

$$\text{sub :} \quad \sum_{l=1}^{L} x_{n,l} = 1, \qquad \forall n, \tag{10}$$

$$x_{n,l} \geq 0, \qquad \forall n, \forall l. \tag{11}$$

Define $\mathcal{Y} = \left\{ \mathbf{y} \,\middle|\, \sum_{l=1}^{L} y_l = 1, \mathbf{y} \geq 0 \right\}$ as the set of all feasible solutions of the LDP satisfying constraints (7) and (8), and let $\mathcal{Y}^* \subseteq \mathcal{Y}$ be the set of all vectors $\mathbf{y} \in \mathcal{Y}$ that are optimal. Also, define $\mathcal{X} = \left\{ \mathbf{x} \,\middle|\, \forall n : \sum_{l=1}^{L} x_{n,l} = 1, \mathbf{x} \geq 0 \right\}$ as the set of all feasible solutions of the LP satisfying constraints (10) and (11), and let $\mathcal{X}^*(\mathbf{y}) \subseteq \mathcal{X}$ be the set of all vectors $\mathbf{x} \in \mathcal{X}$ that are optimal with respect to a given vector $\mathbf{y}$.

For a fixed vector $\mathbf{y}$, the sub-problem (9)-(11) amounts to a minimization of the sum of weighted loads and is trivial to solve. Indeed, an assignment vector $\mathbf{x}(\mathbf{y})$ is optimal with respect to $\mathbf{y}$ if and only if it satisfies the relationship

$$x_{n,l}(\mathbf{y}) > 0 \Rightarrow l \in \arg\min_{l'} \{ y_{l'} w_{l'} / R_{n,l'} \}, \tag{12}$$

or, equivalently,

$$y_l w_l / R_{n,l} > \min_{l'} \{ y_{l'} w_{l'} / R_{n,l'} \} \Rightarrow x_{n,l}(\mathbf{y}) = 0. \tag{13}$$

Any optimal solution $\mathbf{x}^*(\mathbf{y})$ to this sub-problem by definition minimizes a weighted sum of the loads and hence must be Pareto-optimal, assuming $\mathbf{y} > 0$. Conversely, note that the set of achievable load vectors $\boldsymbol{\rho}$ is convex, and hence any Pareto-optimal vector $\mathbf{x}$ must minimize some weighted combination of the loads, and thus must satisfy the structural properties (12) and (13) for some vector $\mathbf{y} \geq 0$. The properties (12) and (13) play a crucial role in the assignment rule of the algorithm introduced in the next section.

IV. Online Algorithm

In this section we describe the Shadow Price Assignment algorithm, or SPA-algorithm, which uses iterated control parameters $\mathbf{y}^{(i)} = (y_1^{(i)}, \ldots, y_L^{(i)})$ related to the dual variables introduced in the previous section. The dual variables are commonly referred to as shadow prices which explains the name of the algorithm. The SPA-algorithm updates the control parameters $\mathbf{y}^{(i)}$ at the start of each file transfer. In the next section we exploit stochastic approximation techniques [20] to establish convergence of the control parameters.

A. Algorithm specification

When a user $i+1$ initiates a file transfer, the SPA-algorithm updates the vector $\mathbf{y}^{(i)}$ based on load measurements. After the update the algorithm uses the vector $\mathbf{y}^{(i+1)}$, in conjunction with physical rate information, to assign the user to an AP. The load measurements (proxies) are determined as follows. Suppose that user i arrives at location n, is assigned to AP $\tilde{l}$, and has a file of size $B^{(i)}$. Define

$$\sigma_0^{(i)} = \frac{B^{(i)}}{R_{n,\tilde{l}}} \tag{14}$$

as the instantaneous load that user i brings to AP $\tilde{l}$, and let $\sigma_l^{(i)} = \sigma_0^{(i)} \mathbb{1}_{\{l=\tilde{l}\}}$. Furthermore, define $W = \sum_{l=1}^{L} 1/w_l$. Then the SPA-algorithm is described as follows, where $\varepsilon > 0$ determines the magnitude of the updates (or step sizes) and is typically small.

SPA-algorithm	
Initialization:	Set $y_l^{(0)} = 1/L$, $\sigma_l^{(0)} = 0$, $\forall l = 1, \ldots, L$.
Update Step:	Update $y_l^{(i+1)} = y_l^{(i)} + \varepsilon \left(\sigma_l^{(i)} - \frac{\sigma_0^{(i)}}{w_l W} \right)$, $\forall l = 1, \ldots, L$.
Assignment Step:	User $i+1$ at location n is assigned to an AP $l \in \arg\min_{l'} \{ w_{l'} y_{l'}^{(i+1)} / R_{n,l'} \}$, where $R_{n,l'} > 0$.

The SPA-algorithm has a similar structure as the user association rules proposed in [21], which also assign a user to an AP that offers the maximum ratio of the physical transmission rate and a load-related variable. In [21], the load-related quantity is obtained by taking an arbitrary negative power of the long-term average fraction of time that the AP is inactive.

B. Interpretation

The update rule increases the control parameter y_l with the local incoming load, and decreases it with a fraction $1/L$ of the global incoming load. Thus the update rule tends to increase/decrease the control parameter y_l for APs l with relatively high/low observed load values. The assignment rule assigns a user at location n to an AP l with a relatively high physical transmission rate $R_{n,l}$ or a relatively low control parameter y_l, consistent with the property in Equation (12). Together, these two components will drive the control parameters y_l to values that maximize $V^*(\mathbf{y})$ as defined in the previous section, and thus allow for the weighted loads to be balanced, while assigning users in a Pareto-optimal fashion. Indeed, in Section V we prove that, under mild assumptions, the sequence $\{\mathbf{y}^{(i)}\}_{i \geq 0}$ realized under the SPA-algorithm converges (in some appropriate sense) to optimal values $\mathbf{y}^* \in \mathcal{Y}^*$.

Remark 3: When the y_l values are not updated but kept fixed and equal at $1/L$, we obtain an assignment strategy which selects the AP l with the highest value for $R_{n,l}/w_l$ at location n, irrespective of the load conditions. This minimizes the sum of the weighted loads, and in particular the aggregate load in case all weights are equal, i.e. $w_l = 1$ for all $l = 1, \ldots, L$. The latter case corresponds to the conventional procedure of choosing the AP with the best SINR, which will be referred to as the Best-SINR algorithm.

Remark 4: Observe that the SPA-algorithm does not explicitly use the discrete set of locations. We could allow a continuum of locations and still apply the SPA-algorithm as long as it is possible to determine the physical rates $R_{i,l}$ which an individual user i can receive from AP l. This can be done by means of SINR measurements, which is already common in current systems.

C. Implementation aspects

To perform the assignment step for user $i+1$ at location n, the algorithm relies on values for $w_l y_l^{(i+1)}$ and $R_{n,l}$. In an actual system implementation, the update step can be decoupled from the assignment decision. Specifically, the updates could be performed asynchronously, independently of file transfer initiations, and the assignment decisions could simply be made using the most recently calculated control parameters.

An alternative option is to perform the updates by using the resource utilizations of the APs rather than the file size $B^{(i)}$ of each user. E.g. proxies of the form $\sigma_l^{(i)} \in [0, 1]$, where $\sigma_l^{(i)}$ is the fraction of resources of AP l being used at the time at which user i initiates a file transfer. The advantage is that this does not require knowledge of the file size of each user, but now we have to determine the load status per AP. Furthermore, this method measures the carried traffic and not the offered traffic, and we should therefore compensate for possible service denials if we wish infer the offered load from the carried load.

As an alternative to the additive update rule, we can implement a multiplicative update rule given by

$$\log \left(y_l^{(i+1)} \right) = \log \left(y_l^{(i)} \right) + \varepsilon \left(\sigma_l^{(i)} - \frac{\sigma_0^{(i)}}{w_l W} \right). \quad (15)$$

The advantage of using a multiplicative update rule like (15) is that the values of $y_l^{(i)}$ are guaranteed to remain positive and it is more in line with the multiplicative nature of the assignment rule. We lose the property that all the values of $y_l^{(i)}$ add up to one, but the sum of the logarithms is kept fixed. In the end the ratios between the values of $y_l^{(i)}$ are what matters, since they determine the association rule. In Section VI we present simulation results for a multiplicative implementation, which also demonstrate convergence.

V. Convergence proof

In this section we prove the convergence of the control parameters $\mathbf{y}^{(i)}$ under the SPA-algorithm with an additive update rule. We assume Poisson arrivals and exponential file size distributions. We also assume that the optimal solution $\mathbf{y}^*$ to the LDP (6)-(8) is strictly positive and unique. In Appendix A we will explain why the latter assumption is reasonable in the present context and demonstrate that it also implies that $\mathbf{y}^*$ in fact optimizes $V(\mathbf{y})$ over all $\mathbf{y} \in \mathcal{Y}_1 = \{\mathbf{y} \in \mathbb{R}^L \mid \sum_{l=1}^L y_l = 1\}$, in particular including vectors with negative components.

We use the framework of Kushner and Yin [20] which implies that under appropriate technical conditions the control parameters $\mathbf{y}^{(i)}$ move to a small neighbourhood around the limit set of a certain differential inclusion (DI). The proof thus involves three steps. Step (i) is to verify all technical conditions in order to apply the framework of Kushner and Yin. Step (ii) is then to identify the relevant DI. Step (iii) is to construct a suitable Lyapunov function which shows that the limit set of the DI consists of the vector $\mathbf{y}^*$.

For notational convenience, we henceforth assume without essential loss of generality that $w_l = 1$ for all APs l, and define $\mathcal{N}_\delta(\mathbf{y}^*)$ as a δ-neighbourhood around $\mathbf{y}^*$. Then, the control parameters under the SPA-algorithm converge to the vector $\mathbf{y}^*$ in the sense described by the following theorem.

Theorem 1: For any $\delta > 0$, the fraction of time that the sequence of control parameters $\{\mathbf{y}_\varepsilon^{(i)}\}_{\{0 \le i \le I/\varepsilon\}}$ produced by the SPA-algorithm with fixed step size ε spends in $\mathcal{N}_\delta(\mathbf{y}^*)$ goes to one (in probability) as $\varepsilon \downarrow 0$ and $I \to \infty$.

Step (i): As mentioned above, the first step is verifying that all technical conditions for the framework of Kushner and Yin to apply are satisfied under Markovian traffic assumptions. This is a substantial effort, but not difficult, and the details are omitted due to space limitations. The framework of Kushner and Yin [20, Theorem. 8.2.5] (henceforth referred to as the KY-Theorem) then implies that the behaviour of the sequence $\{\mathbf{y}_\varepsilon^{(i)}\}_{\{0 \le i \le I/\varepsilon\}}$ produced by the SPA-algorithm, when suitably rescaled, is characterized by a DI which can be written as

$$\frac{d}{dt}\mathbf{y}(t) \in \mathcal{G}(\mathbf{y}(t)), \tag{16}$$

where $\mathcal{G}(\mathbf{y}(t))$ is a set that will be specified in Step (ii), $\mathbf{y}(0) = (1/L)\vec{1}_L$, and $\vec{1}_L$ a vector of length L containing only ones. Furthermore, the KY-theorem states the following. Let $LPS_\mathcal{G}$ (Limit Point Set) be the set of limit points of the DI (16) and let $\{\mathbf{y}_\varepsilon^{(i)}\}_{\{0 \le i \le I/\varepsilon\}}$ be a sequence of control vectors generated by the SPA-algorithm with update size ε. Then, for any $\delta > 0$, the fraction of time that $\{\mathbf{y}_\varepsilon^{(i)}\}_{\{0 \le i \le I/\varepsilon\}}$ spends in $\mathcal{N}_\delta(LPS_\mathcal{G})$ goes to one (in probability) as $\varepsilon \downarrow 0$ and $I \to \infty$. This result already brings us close to the statement of Theorem 1. It remains to determine the set $\mathcal{G}(\mathbf{y})$ (Step (ii)) and then establish that (16) implies that $\mathbf{y}(t) \to \mathbf{y}^*$, i.e. $LPS_\mathcal{G} = \{\mathbf{y}^*\}$.

Step (ii): In order to determine the set $\mathcal{G}(\mathbf{y})$ in the DI, it is convenient to introduce $g^{(i)} = \boldsymbol{\sigma}^{(i)} - (\sigma_0^{(i)}/L)\vec{1}_L$ as the update direction in the i-th iteration of the SPA-algorithm, with $\boldsymbol{\sigma}^{(i)} = (\sigma_1^{(i)}, \ldots, \sigma_L^{(i)})$.

Informally speaking, when $\mathbf{y}(t) = \mathbf{y}$, the KY-theorem states that the time derivative $\frac{d}{dt}\mathbf{y}(t)$ is determined by the expectation $h(\mathbf{y})$ of the $g^{(i)}$ values in a system where the control parameters are held fixed at $\mathbf{y}$ at all times, provided the function $h(\mathbf{y})$ is continuous. In such a system, the assignment fractions $x_{n,l}$ must satisfy the structural properties in Equations (12) and (13), and thus the function $h(\mathbf{y})$ would be of the form $h(\mathbf{y}) = \boldsymbol{\rho}(\mathbf{x}(\mathbf{y})) - \rho_0(\mathbf{x}(\mathbf{y}))\vec{1}_L$, with $\rho_0(\cdot) = (1/L)\sum_{l=1}^{L}\rho_l(\cdot)$. However, $\boldsymbol{\rho}(\mathbf{x}(\mathbf{y}))$ is not uniquely determined and not continuous, and hence the function $h(\mathbf{y})$ is not uniquely determined, and not continuous either. Therefore, additional care is needed, and rather than an ordinary differential equation, the behaviour of $\mathbf{y}(t)$ is described by the DI (16), where the set $\mathcal{G}(\mathbf{y})$ is such that for any sequence $\mathbf{y}^{(i)}$, $\mathbf{y}$, satisfying $\lim_{k,k'\to\infty} \sup_{k \le i \le k+k'} \left|\mathbf{y}^{(i)} - \mathbf{y}\right| = 0$ we have

$$\lim_{k,k'\to\infty} d\left(\frac{1}{k'}\sum_{i=k}^{k+k'-1}\mathbb{E}\left[g^{(i)}\right], \mathcal{G}(\mathbf{y})\right) = 0, \tag{17}$$

where $d(s, S)$ measures the distance of an element s to a set S. In other words, for any sequence of vectors $\{\mathbf{y}^{(i)}\}_{i\ge1}$ converging to a vector $\mathbf{y}$, the long-term average value of $g^{(i)}$ lies in the set $\mathcal{G}(\mathbf{y})$. It can be checked that this holds for the set $\mathcal{G}(\mathbf{y})$ defined by

$$\mathcal{G}(\mathbf{y}) = \left\{\boldsymbol{\rho}(\mathbf{x}) - \rho_0(\mathbf{x})\vec{1}_L \,\middle|\, \mathbf{x} \in \mathcal{X}^*(\mathbf{y})\right\}, \tag{18}$$

with $\mathcal{X}^*(\mathbf{y})$ as defined in Section III.

Step (iii): Now that we have identified the DI we focus on the third step: constructing a Lyapunov function to establish that $LPS_\mathcal{G} = \{\mathbf{y}^*\}$. Consider the candidate function given by

$$D(\mathbf{y}) = \sum_{l=1}^{L}(y_l - y_l^*)^2. \tag{19}$$

By the uniqueness of $\mathbf{y}^*$, we can derive the following result.

Lemma 1: The function $D(\mathbf{y})$ given by (19) satisfies $\frac{d}{dt}D(\mathbf{y}(t)) < 0$ as long as $\mathbf{y}(t) \neq \mathbf{y}^*$ and $\mathbf{y}(t) \in \mathcal{Y}_1$.

The proof is given in Appendix B. By Lemma 1 we know that the function $D(\mathbf{y}(t))$ is a Lyapunov function for the DI given by (16) on the set $\mathcal{Y}_1$ and hence $\mathbf{y}^*$ is locally asymptotically stable in $\mathcal{Y}_1$, i.e. $LPS_\mathcal{G} = \{\mathbf{y}^*\}$.

Remark 5: Theorem 1 applies to situations where ε is kept fixed while running the SPA-algorithm. With a variable step size $\varepsilon^{(i)}$ satisfying $\varepsilon^{(i)} \downarrow 0$, $\sum_{i=1}^{\infty}\varepsilon^{(i)} = \infty$ and $\sum_{i=1}^{\infty}(\varepsilon^{(i)})^2 < \infty$ (e.g. $\varepsilon^{(i)} = 1/i$), we can obtain a stronger form of convergence, implying that $\mathbf{y}^{(i)} \to \mathbf{y}^*$ as $i \to \infty$ almost surely [20, Theorem 8.2.3]. However, for practical applications it is desirable to keep ε small but fixed to ensure that the algorithm can respond to changes in system parameters in non-stationary scenarios.

Remark 6: In the end the optimal values of $\mathbf{y}$ are a means to realize optimal assignments $\mathbf{x}^*$ without knowing values of λ_n and β_n. Let $x_{n,l}^{(i)}$ be the fraction of users at location n up to and including user i which have been assigned to AP l by the SPA-algorithm. Results of Sherali and Choi [22] imply that, as $\mathbf{y}^{(i)} \to \mathbf{y}^*$, $\mathbf{x}^{(i)} \to \mathbf{x}^*$ as $i \to \infty$, where $\mathbf{x}^*$ is an optimal solution to (2)-(5). In other words, the SPA-algorithm realizes optimal assignments in the long run.

Remark 7: In Section IV-C we considered allowing a continuum of user locations as this would better suit practical situations. In addition, it makes the loads $\rho_l(\mathbf{x}(\mathbf{y}))$ continuous as functions of $\mathbf{y}$ and so we would obtain an ordinary differential equation rather than a DI. However, it also requires an infinite-dimensional state space to describe the system state. This poses severe complications in verifying some of the technical conditions needed to apply the framework of Kushner and Yin.

VI. Simulation results

We now discuss several results of simulation experiments which we conducted to gain insight in the performance of the proposed SPA-algorithm. We used the SPA-algorithm with the multiplicative update rule as given in (15), using various step sizes: $\varepsilon_\downarrow^{(i)} = 1/(i+1)$ (decreasing), $\varepsilon_{\downarrow s}^{(i)} = (1/(i+1))^{2/3}$ (decreasing slower), $\varepsilon_4 = 10^{-4}$ (fixed), $\varepsilon_5 = 10^{-5}$ (fixed), and $\varepsilon_6 = 10^{-6}$ (fixed). Moreover, we calculated the updates by using AP

utilization proxies as explained in Section IV-C. We in particular make comparisons with two benchmark algorithms. The first one is the Best-SINR algorithm introduced in Section IV. The second benchmark algorithm we will refer to as the BIR (Best Instantaneous Rate) algorithm [23], and works as follows. Define $m_l^{(i)}$ as the number of users being served by AP l just before the arrival of user i. User i will be assigned to AP $l \in \arg\min_{1\le l'\le L}\{(m_l^{(i)}+1)/R_{n,l}\}$. This assignment rule has the same structure as that of the SPA-algorithm with a state-dependent variable $m_l^{(i)}$ instead of a control parameter $y_l^{(i)}$. Basically, each user is assigned to the AP which offers the maximum instantaneous throughput it can expect directly after being assigned.

We simulated two scenarios, both a $1500 \times 1500m$ area with 63 APs and having some hotspot zone(s). The two scenarios are shown combined in Figure 1. To

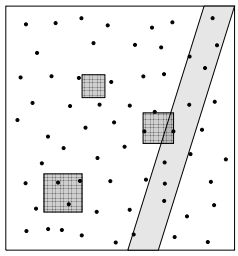


Figure 1: The black discs represent APs. Scenario 1 has hotspots marked by the dotted squares. Scenario 2 has a hotspot marked by the grey area.

better represent realistic scenarios, we used a continuum of locations in both scenarios and for the achievable physical transmission rates we used the 3GPP TR 36.814 V9.0.0. urban-micro cell pathloss model: the APs have orthogonal spectrum resources with a bandwidth of 5 MHz, they transmit with a power of 30 dBm and the thermal noise is -174 dBm. The resulting pathloss factor is then $140.7 + 36.7\log_{10}(d_{i,l}/1000)$ dB, where $d_{i,l}$ is the distance in meters from the location of user i to AP l. A single AP can admit at most 100 users simultaneously and all APs fairly share the resources among all active users. A user is blocked (service denial) when it is assigned to an AP which already has 100 active users. Moreover, to reduce boundary effects, we glued the left and right boundary together.

The load balancing capability is particularly critical in scenarios where the baseline Best-SINR algorithm results in structurally overloaded APs, while a load-aware assignment strategy can avoid this. We chose the following simulation values. For Scenario 1, the aggregate intensity of file transfer initiations in the complete area is $\lambda = 20$ per second, and the relative intensities of the complete remaining area and the hotspots (dotted squares, in top to bottom order) are $1 : 15 : 10 : 8$. Moreover, the mean file size is $\beta = 6$ Mb, independent of the location. For Scenario 2, the aggregate intensity of file transfer initiations is also $\lambda = 20$ per second, but the intensity in the diagonal hot spot band is three times higher than the remaining part. In this case, we have $\beta = 4.5$ Mb. The values are such that the BIR algorithm admits all users (no service denials), while the Best-SINR results in overloaded APs (which can be recognized by observing service denials).

In Figures 2 and 3 we see the evolution of one control parameter $y_3^{(i)}$ of a single run, when using three different step sizes: ε_4, $\varepsilon_\downarrow^{(i)}$, and $\varepsilon_{\downarrow s}^{(i)}$. We focus on AP 3 since in both Scenarios 1 and 2 it shows service denials under the Best-SINR algorithm, while the SPA and BIR algorithms can avoid this (see Table I). Clearly we see that the control parameters with ε_4 and $\varepsilon_{\downarrow s}^{(i)}$ evolve (with variability) around an optimal value. Indeed, the variability with $\varepsilon_{\downarrow s}^{(i)}$ is decreasing and the control parameter converges. With $\varepsilon_\downarrow^{(i)}$ the control parameter does not visibly converge as the update steps are decreasing too fast. The step sizes ε_5 and ε_6 are left out as they would appear as horizontal lines on the scale of Figures 2 and 3 due to the small step sizes. In short, we clearly observe a trade-off between convergence speed and accuracy. Small step sizes show slow convergence but less variability (higher accuracy), while larger step sizes move to optimal values faster but also cause a more erratic evolution.

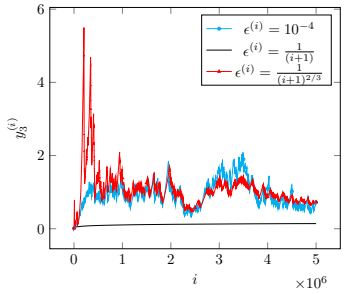


Figure 2: Evolution of $y_3^{(i)}$ in Scenario 1, using various step sizes.

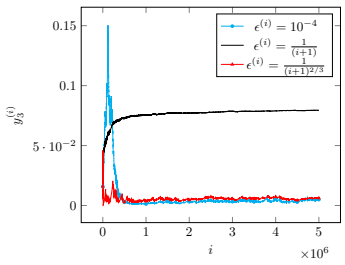


Figure 3: Evolution of $y_3^{(i)}$ in Scenario 2, using various step sizes.

The choice of step sizes ε or $\varepsilon^{(i)}$ not only influences the evolution of the control parameters, but also the performance of the SPA-algorithm. In Table I we present the fractions of service denials of users 900 000 to 1 000 000 in a single simulation run. The algorithms labelled as 'Other' are the SPA-algorithm with step sizes $\varepsilon_{\downarrow s}^{(i)}$, ε_4, and ε_5, and

AP	Best-SINR S1	Best-SINR S2	SPA ε_6 S1	SPA ε_6 S2	SPA $\varepsilon_\downarrow$ S1	SPA $\varepsilon_\downarrow$ S2	Other S1 & S2
3	0.215	0.140	0.152	0.004	0	0	0
5	0.165	0.033	0.050	0	0	0	0
11	0.080	0	0	0	0	0	0
12	0.077	0	0.013	0	0	0	0
14	0	0	0	0	0.012	0	0
24	0	0	0	0	0	0	0
25	0.271	0	0.244	0	0	0	0
34	0.212	0	0.144	0	0	0	0
37	0	0.248	0	0.221	0	0	0
38	0	0.170	0	0	0	0	0
46	0	0.136	0	0.114	0	0	0
50	0.067	0	0	0	0	0	0
51	0	0.146	0	0.005	0	0	0
60	0.128	0.190	0.012	0.169	0	0	0
61	0	0.255	0	0.172	0	0	0

Table I: Blocking fractions at APs in Scenario 1 (S1) and Scenario 2 (S2), only listing APs with service denials.

the BIR algorithm. For all choices of the step size ε or $\varepsilon^{(i)}$, the SPA-algorithm realizes significantly fewer service denials than the Best-SINR algorithm, and with many step sizes it shuns them completely: only the smallest step sizes, $\varepsilon_\downarrow^{(i)}$ and ε_6 still show service denials. We see that the SPA-algorithm can indeed recognize overloaded APs and redirect load such that all users are served.

The user-perceived throughput is defined as the user's file size divided by its total serving time. Without overloaded APs we see an improvement in user-perceived throughput for users with low throughput values, for which it is the most critical. Indeed, a throughput value below a certain threshold may be interpreted as a "soft" service outage, and is a critical KPI to service providers. In Figures 4 to 7 we plotted the user-perceived throughput of various groups of users in the two scenarios, again using the different step sizes. We omitted the plot for $\varepsilon_{\downarrow,s}^{(i)}$ since it is very close to that of ε_4: around users 900 000 to 1 000 000 they have approximately equal step sizes.

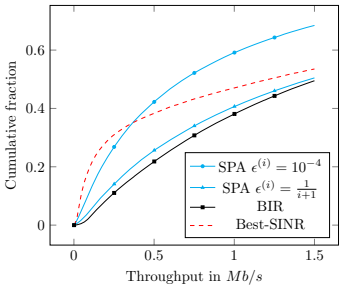


Figure 4: Empirical cumulative distributions of the throughput for users 900 000 to 1 000 000, for various algorithms in Scenario 1.

Observe that larger values of ε are too crude to realize (close to) optimal ratios for the y_l values. That means that with ε_4 the SPA-algorithm is not performing optimally with respect to throughput. On the other hand, smaller values of ε are slower in convergence, but the relative ratios between the values are better maintained, which

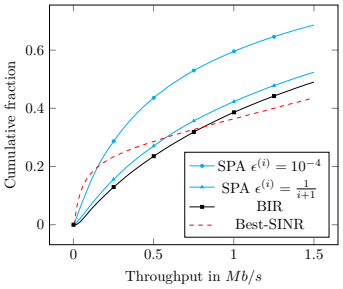
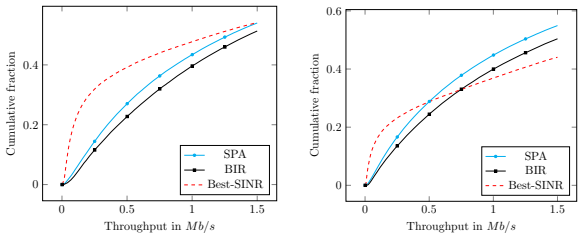


Figure 5: Empirical cumulative distributions of the throughput for users 900 000 to 1 000 000, for various algorithms in Scenario 2.

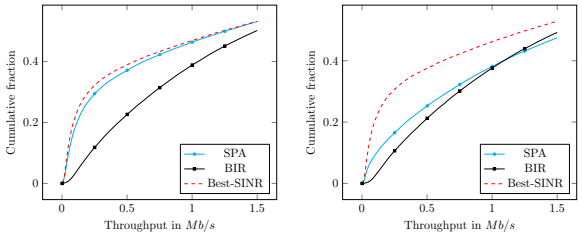


(a) Scenario 1. (b) Scenario 2.

Figure 6: Empirical cumulative distributions of the throughput for users Users 900 000 to 1 000 000, with $\varepsilon = 10^{-5}$.

translates to better a throughput performance. This can be seen by combining Table I and Figure 7: we can see that the throughput performance has improved tremendously over time even though the algorithm still shows service denials, meaning it has not converged sufficiently yet. Using ε_5 or $\varepsilon_\downarrow^{(i)}$ both results in a good throughput performance and (almost) completely shuns service denials. The step size $\varepsilon_\downarrow$ quickly realizes a good performance, yet may be unable to adapt fast enough at a later stage.

We observe that especially for users with low throughput the SPA-algorithm is outperforming the Best-SINR algorithm. For higher throughputs it appears that Best-SINR is overtaking both the BIR and the SPA-algorithm, but at the expense of many service denials. Indeed, the



(a) Users 400 000 to 500 000. (b) Users 900 000 to 1 000 000.

Figure 7: Empirical cumulative distributions of the throughput for some selected users in Scenaro 1, with $\varepsilon = 10^{-6}$.

SPA-algorithm and the BIR-algorithm identify overloaded APs and realize a stable user assignment when possible. It is then only expected that this goes at the expense of the throughput performance for users with high throughput levels. Moreover, we have presented results for several choices of ε and $\varepsilon^{(i)}$ which show that there is a crucial trade-off between performance and convergence speed.

VII. Conclusion

In this paper we analysed a load-aware (pico-)cell selection algorithm which assigns users to APs based on control parameters. The control parameters are iteratively adjusted by means of load measurements and we proved that under suitable assumptions the iterates converge to optimal values of the control parameters. We demonstrated that in two different scenarios the SPA-algorithm improves the throughput of users with low perceived throughput and realizes a stable user assignment when possible. We also extensively treated the impact of the step sizes on the performance and control parameter evolution and showed that there is a trade-off between convergence speed, throughput performance, and service denials. Rigorously proving the convergence of the algorithm with a continuum of user locations or the multiplicative update rule, and considering scenarios where transmission rates are strongly impacted by variations in activity levels of surrounding APs are interesting challenges for future research. Other possible extensions may consider the SPA-algorithm in scenarios where AP functionality is liable to disruptions (self-healing), e.g. technical failures or terrorist attacks, and scenarios where APs can be intentionally activated or de-activated (self-configuring), e.g. for saving energy in areas of low traffic density or installing new APs to provide more capacity in hot spot zones.

Appendix A. Uniqueness of $\mathbf{y}^*$

In Section V we proved that the fraction of the control vectors $\mathbf{y}^{(i)}$ that belong to an arbitrarily small neighbourhood of the optimal vector $\mathbf{y}^*$ tends to one over sufficiently long time intervals when the step size ε becomes correspondingly small, provided $\mathbf{y}^*$ is unique and strictly positive. We now demonstrate that in most settings of interest the latter assumption is reasonable.

For a given user assignment $\mathbf{x}$, let $G(\mathbf{x})$ be a (bipartite) graph with vertex set $V = \{1,\ldots,N\} \cup \{1,\ldots,L\}$ and edge set $E = \{(n,l) \in \{1,\ldots,N\} \times \{1,\ldots,L\} : x_{n,l} > 0\}$. It can be argued that in typical scenarios, the graph $G(\mathbf{x}^*)$ associated with any optimal user assignment $\mathbf{x}^*$ is connected. Indeed, if that were not the case, then under the optimal user assignment the wireless network would not be connected in the sense that it would operate as (at least) two disjoint components. More specifically, in that case the set of user locations could be partitioned into $C \geq 2$ non-empty subsets $\mathcal{N}_1,\ldots,\mathcal{N}_C$ and the set of APs could be partitioned into $C \geq 2$ subsets $\mathcal{L}_1,\ldots,\mathcal{L}_C$ such that all user locations in $\mathcal{N}_c$ are assigned to APs in $\mathcal{L}_c$, $c = 1,\ldots,C$. In other words, no load is shared across the various network components, and in particular the minimum achievable maximum load is no less than when the maximum load is minimized within each of the components individually. Hence, the load balancing problem can basically be tackled within each of the components separately, and we conclude that in case the graph $G(\mathbf{x}^*)$ is not connected, the global load balancing problem can essentially be reduced to independent sub-problems.

Based on the above, in most relevant situations the graph $G(\mathbf{x}^*)$ associated with any optimal user assignment $\mathbf{x}^*$ must be connected, and we henceforth assume that to be the case. We will show that this implies that the solution to the optimization problem (6)-(7) is strictly positive and unique, even in the absence of the non-negativity conditions (8). First, suppose that there exists an optimal solution $\mathbf{y}^*$ that is not strictly positive. Let $\mathcal{K} = \{k \in \{1,\ldots,L\} : y_k^* \leq 0\} \neq \emptyset$, and observe that $\mathcal{K} \neq \{1,\ldots,L\}$ since $\sum_{l=1}^{L} y_l^* = 1$. Further, introduce $\mathcal{M} = \{n \in \{1,\ldots,N\} : \max_{k\in\mathcal{K}} R_{n,k} > 0\}$. In view of properties (12) and (13), any optimal user assignment vector $\mathbf{x}^*$ must satisfy $x_{n,k}^* = 0$ for all $k \in \mathcal{K}$, $n \notin \mathcal{M}$ and $x_{m,l}^* = 0$ for all $l \notin \mathcal{K}$, $m \in \mathcal{M}$, so that the graph $G(\mathbf{x}^*)$ is disconnected, which yields a contradiction. Because of the complementary slackness conditions, the fact that $\mathbf{y}^*$ is strictly positive also means that the loads are strictly balanced under the optimal user assignment, i.e., $\sum_{n=1}^{N} x_{n,l}^* \lambda_n \beta_n / R_{n,l} = U^*$ for all $l = 1,\ldots,L$.

Now suppose that the solution to the optimization problem (6)-(8) is not unique, i.e., there are two different solutions $\mathbf{y}^* \neq \mathbf{z}$, both strictly positive. Define $\alpha = \min_{l=1,\ldots,L} z_l/y_l^* < 1$, and denote $\mathcal{K} = \{k \in \{1,\ldots,L\} : z_k/y_k^* = \alpha\}$. Let $\mathbf{x}^y$ and $\mathbf{x}^z$ be the optimal associated user assignment vectors. For brevity, introduce $x_{n,\mathcal{K}}^y = \sum_{k\in\mathcal{K}} x_{n,k}^y$, and further let $\mathcal{M} = \{n \in \{1,\ldots,N\} : x_{n,\mathcal{K}}^y > 0\}$. In view of the optimality property (12) we have $\max_{k\in\mathcal{K}} R_{n,k}/y_k^* \geq \max_{l\notin\mathcal{K}} R_{n,l}/y_l^*$, $\forall\ n \in \mathcal{M}$. Noting that $z_k/y_k^* < z_l/y_l^*$ for all $k \in \mathcal{K}$ and $l \notin \mathcal{K}$, it follows that $\max_{k\in\mathcal{K}} R_{n,k}/z_k > \max_{l\notin\mathcal{K}} R_{n,l}/z_l$, $\forall\ n \in \mathcal{M}$. The optimality property (12) then implies $x_{n,\mathcal{K}}^z = 1$ for all $n \in \mathcal{M}$. Since the graph $G(\mathbf{x}^y)$ is connected, there must exist APs $k \in \mathcal{K}$ and $l \notin \mathcal{K}$ and a user location n such that $x_{n,k}^y > 0$, i.e., $n \in \mathcal{K}$, and $x_{n,l}^y > 0$, so that $x_{n,\mathcal{K}}^y < 1$. This means that the aggregate load of the APs in $\mathcal{K}$ under the user assignment $\mathbf{x}^z$ must be strictly larger than under the user assignment $\mathbf{x}^y$:

$$\sum_{k\in\mathcal{K}} \sum_{n=1}^{N} x_{n,k}^z \lambda_n \beta_n / R_{n,k} > \sum_{k\in\mathcal{K}} \sum_{n=1}^{N} x_{n,k}^y \lambda_n \beta_n / R_{n,k}. \tag{20}$$

As mentioned above, the loads are strictly balanced under the optimal user assignment $\mathbf{x}^y$, i.e., $\sum_{n=1}^{N} x_{n,l}^y \lambda_n \beta_n / R_{n,l} = U^*$ for all $l = 1,\ldots,L$, and in particular $\sum_{n=1}^{N} x_{n,k}^y \lambda_n \beta_n / R_{n,k} = U^*$ for all $k \in \mathcal{K}$. Inequality (20) then implies that the load at one of the APs under the user assignment x^z must be strictly larger, $\max_{k\in\mathcal{K}} \{\sum_{n=1}^{N} x_{n,k}^z \lambda_n \beta_n / R_{n,k}\} > U^*$, which yields a contradiction with the presumed optimality of the user

assignment $\mathbf{x}^z$. We conclude that the solution of the optimization problem (6)-(8) must be unique.

APPENDIX B. PROOF OF LEMMA 1

Using the DI given by (16) we write $\frac{d}{dt}y_l(t) = \Delta_l(\mathbf{y}(t))$, and the time-derivative of $D(\mathbf{y}(t))$ can be written as:

$$\frac{d}{dt}D(\mathbf{y}(t)) = 2\sum_{l=1}^{L}(y_l(t) - y_l^*)\Delta_l(\mathbf{y}(t)), \qquad (21)$$

with $(\Delta_1(\mathbf{y}(t)), \ldots, \Delta_L(\mathbf{y}(t))) \in \mathcal{G}(\mathbf{y}(t))$ and $\Delta_l(\mathbf{y}(t)) = \rho_l(\mathbf{x}(\mathbf{y}(t)) - \rho_0(\mathbf{x}(\mathbf{y}(t))$ for some $\mathbf{x}(\mathbf{y}(t)) \in \mathcal{X}^*(\mathbf{y}(t))$. Then, for all $\mathbf{x}(\mathbf{y}(t)) \in \mathcal{X}^*(\mathbf{y}(t))$ with $\mathbf{y}(t) \neq \mathbf{y}^*$, $\frac{d}{dt}D(\mathbf{y}(t))$ satisfies:

$$(21) = 2\left(\sum_{l=1}^{L} y_l(t)\rho_l(\mathbf{x}(\mathbf{y}(t))) - \sum_{l=1}^{L} y_l^*\rho_l(\mathbf{x}(\mathbf{y}(t)))\right) \qquad (22)$$

$$= 2\left(V^*(\mathbf{y}(t)) - V^*(\mathbf{y}^*)\right) \qquad (23)$$

$$+ 2\left(V^*(\mathbf{y}^*) - \sum_{l=1}^{L} y_l^*\rho_l(\mathbf{x}(\mathbf{y}(t)))\right) \qquad (24)$$

$$< 0.$$

To obtain (23) + (24) from (22) we use that $\sum_{l=1}^{L} y_l^* = 1$, and add and subtract $V^*(\mathbf{y}^*)$. Also, since $\mathbf{y}(0) = (1/L)\vec{1}_L \in \mathcal{Y}$ and $\sum_{l=1}^{L}\Delta_l(\mathbf{y}(t)) = 0$ we know $\sum_{l=1}^{L} y_l(t) = 1$. The expression in (23) is thus strictly negative since $\mathbf{y}^*$ uniquely maximizes $V^*(\mathbf{y})$ over $\mathbf{y} \in \mathcal{Y}_1$ and we assumed that $\mathbf{y}(t) \neq \mathbf{y}^*$. The second part, given by (24), is non-positive because $\boldsymbol{\rho}(\mathbf{x}(\mathbf{y}))$ minimizes $V(\mathbf{y})$, and in particular $\boldsymbol{\rho}(\mathbf{x}(\mathbf{y}^*))$ minimizes $V(\mathbf{y}^*)$.

REFERENCES

[1] A. M. Koonen, M. G. Larrode, A. Ng'Oma, K. Wang, H. Yang, Y. Zheng, and E. Tangdiongga, "Perspectives of radio-over-fiber technologies," in *Conf. Opt. Fiber Comm./Nat. Fiber Opt. Eng. Conf.* IEEE, 2008.

[2] M. Sauer, A. Kobyakov, and J. George, "Radio over fiber for picocellular network architectures," *J. Lightwave Tech.*, vol. 25, no. 11, pp. 3301–3320, 2007.

[3] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, "Femtocells: Past, present, and future," *IEEE J. Sel. Areas Comm.*, vol. 30, no. 3, pp. 497–508, 2012.

[4] J. G. Andrews, "Seven ways that HetNets are a cellular paradigm shift," *IEEE Comm. Mag.*, vol. 51, no. 3, pp. 136–144, 2013.

[5] B. Post and S. Borst, "Optimal cell assignment algorithms for pico-cell networks," in *Proc. MASCOTS 2016*. IEEE, 2016, pp. 343–348.

[6] A. L. Stolyar, "Optimal routing in output-queued flexible server systems," *Prob. Eng. Inf. Sci.*, vol. 19, no. 02, pp. 141–189, 2005.

[7] S. Borst, G. Hampel, I. Saniee, and P. Whiting, "Load balancing in cellular wireless networks," in *Handbook of Optimization in Telecommunications*. Springer, 2005, pp. 941–978.

[8] S. Borst, I. Saniee, and P. Whiting, "Distributed dynamic load balancing in wireless networks," *Lecture Notes in Computer Science*, 2007.

[9] R. Ramjee, T. Bu, and L. E. Li, "Generalized proportional fair scheduling in third generation wireless data networks," in *Proc. INFOCOM 2006*. IEEE, 2006.

[10] R. Chai, H. Zhang, X. Dong, Q. Chen, and T. Svensson, "Optimal joint utility based load balancing algorithm for heterogeneous wireless networks," *Wireless Networks*, vol. 20, no. 6, pp. 1557–1571, 2014.

[11] W. Li, Y. Cui, S. Wang, and X. Cheng, "Approximate optimization for proportional fair AP association in multi-rate WLANs," in *Wireless Algorithms, Systems, and Applications*. Springer, 2010, pp. 36–46.

[12] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *Proc. INFOCOM 2008*. IEEE, 2008.

[13] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Comm.*, vol. 12, no. 6, pp. 2706–2716, 2013.

[14] H. Zhou, P. Fan, and J. Li, "Global proportional fair scheduling for networks with multiple base stations," *IEEE Trans. Veh. Tech.*, vol. 60, no. 4, pp. 1867–1879, 2011.

[15] F. Moety, M. Bouhtou, T. En-Najjary, and R. Nasri, "Joint optimization of user association and user satisfaction in heterogeneous cellular networks," in *ITC 28*, vol. 1. IEEE, 2016, pp. 78–86.

[16] A. Sankararaman, J.-w. Cho, and F. Baccelli, "Performance-oriented association in large cellular networks with technology diversity," in *ITC 28*, vol. 1. IEEE, 2016, pp. 94–102.

[17] R. Etkin, A. Parekh, and D. Tse, "Spectrum sharing for unlicensed bands," *IEEE J. Sel. Areas Comm.*, vol. 25, no. 3, pp. 517–528, 2007.

[18] D. Shah and J. Shin, "Dynamics in congestion games," in *ACM SIGMETRICS Perf. Eval. Rev.*, vol. 38, no. 1. ACM, 2010, pp. 107–118.

[19] P. Key and A. Proutiere, "Routing games with elastic traffic," *ACM SIGMETRICS Perf. Eval. Rev.*, vol. 37, no. 2, pp. 63–64, 2009.

[20] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer Science & Business Media, 2003, vol. 35.

[21] H. Kim, G. De Veciana, X. Yang, and M. Venkatachalam, "α-optimal user association and cell load balancing in wireless networks," in *Proc. INFOCOM 2010*. IEEE, 2010.

[22] H. D. Sherali and G. Choi, "Recovery of primal solutions when using subgradient optimization methods to solve Lagrangian duals of linear programs," *OR Letters*, vol. 19, no. 3, pp. 105–113, 1996.

[23] S. Borst, A. O. Kaya, D. Calin, and H. Viswanathan, "Optimal path selection in multi-RAT wireless networks," in *Proc. INFOCOM 2016 5G & Beyond Workshop*. IEEE, 2016, pp. 592–597.