Optimizing User Association and Activation Fractions in Heterogeneous Wireless Networks

Vaibhav Singh, Narayan Prasad, Mustafa Y. Arslan, Sampath Rangarajan
e-mail: vaibhavs@umd.edu, {prasad, marslan, sampath}@nec-labs.com

***Abstract*—We consider the problem of maximizing the alpha-fairness utility over the downlink of a heterogeneous wireless network (HetNet) by jointly optimizing *the association* of users to transmission points (TPs) and the *activation fractions* of all TPs. Activation fraction of each TP is the fraction of the frame duration for which it is active, and together these fractions influence the interference seen in the network. To address this joint optimization problem we adopt an approach wherein the activation fractions and the user associations are optimized in an alternating manner. The sub-problem of determining the optimal activation fractions is solved using an auxiliary function method that we show is provably convergent and is amenable to distributed implementation. On the other hand, the sub-problem of determining the user association is solved via a simple combinatorial algorithm. Meaningful performance guarantees are derived and a distributed variant offering identical guarantees is also proposed. The significant benefits of using the proposed algorithms are then demonstrated via realistic simulations.**

I. Introduction

It is well established by now that future cellular networks will be dense HetNets formed by a multitude of disparate transmission points deployed in a highly irregular fashion [1]. In a majority of these deployments, the transmission points (TPs) will be connected to each other by a non-ideal backhaul with a relatively high latency (several dozens of milliseconds). An unfortunate consequence of such a high latency is that it renders unsuitable resource management (RM) schemes that strive to coordinate and obtain allocation decisions within a fine time-scale (e.g., 1 ms in LTE HetNets) [2]–[7]. Instead, semi-static resource management schemes where RM is performed at two time scales, are better suited since they are more robust towards backhaul latency. Broadly speaking, in any such semi-static scheme the RM that is done at a coarse *frame* level granularity (that is at-least as large as the backhaul latency) entails coordination among TPs based on averaged (not instantaneous) slowly varying metrics. On the other hand, the RM in such a scheme that is done at a much finer *slot* level granularity involves no coordination among TPs and is done independently by each active TP based on fast changing metrics [8]–[12]. The semi-static scheme that we propose in this paper decides at the onset of each frame *which set of users should each TP serve over that frame such that each user is served by exactly one TP (user association) and how often should each TP transmit over that frame (activation fraction of that TP)*.

The problem at hand is quite challenging due to the well recognized interference coupling problem. Indeed, while increasing the activation fraction (AF) of a TP will help it serve more users (or serve a given set of users better), it injects more interference to all users being served by other TPs. User Association (without AF optimization) is by itself a popular HetNet RM scheme, wherein the interference coupling problem is simplified by assuming that the interference that would be seen by any user upon being associated to any TP remains static. Association is then determined by optimizing a system utility [13]–[17], or by minimizing a cost function given traffic demands [9], or by adopting a game theoretic framework [18]. Joint optimization of user association along with other system resources, such as power and bandwidth in the downlink [6], [10]–[12], [19] and user powers and TP locations in the uplink [20], [21], has also received significant attention. Considering the downlink which is our focus in this paper, we see that the alternating optimization framework is a popular approach to ensure tractability, and that binary (on-off) power control has been found to be particularly effective in terms of being robust and capturing most of the available gains with a small signalling footprint. The latter observation has led to another promising downlink semi-static RM technique that is fully compliant with the LTE standard, and seeks to capture the benefits of slot-level coordinated binary power control over a HetNet with a non ideal backhaul. This scheme combines user association with partial muting of the high power Macro TP, i.e., the Macro TP is allowed to be active (or transmit with a pre-determined power) for any fraction of the total number of slots in a frame. The choice of this AF for the macro TP is optimized together with the user association [22], [23]. The macro TP then adopts a muting pattern (which includes its on-off status on all the slots) conforming to the determined AF. Notice that the exact on-off status of the macro TP on all the slots is not optimized. Indeed, doing so can be detrimental since coordination done at a coarse time-scale based on the available averaged metrics cannot adapt to the fast changing channel and interference conditions seen across the slots.

Recent studies have shown that topologies without one common dominant interferer will be ubiquitous and in such cases optimizing the AF of only one TP is not enough. The problem we seek to solve is geared exactly towards such deployments. One attempt to solve our problem would be to extend the solutions proposed for the aforementioned scheme, but then it becomes immediately clear that those solutions do not scale when activation fractions for all TPs have to be optimized. This is because those solutions explicitly maintain a rate for each TP-user link under each possible interference pattern, which grow exponentially in the number of TPs. In this paper, we propose a simple formulation that imposes activation fractions and yields one average rate expression for each TP-user link. The latter expression is conservative and is a closed-form function of all activation fractions. Interestingly, in the absence of fast fading our rate expression reduces to the approximate rate expression introduced in [24] (see also [25]), which considered the problem of determining activation fractions to meet a given set of user traffic demands for a

978-3-9018-8273-9/15/ ©2015 IFIP

given user association. We confirm the observation made in those works that the rate expression is in-fact quite accurate over practical HetNets. Our main contributions are as follows:

- We adopt $\alpha-$fairness utility as the system wide utility which generalizes all popular utility functions [26], wherein we also allow for assigning any arbitrary set of weights (reflecting priorities) to the users. We develop centralized and distributed algorithms that yield good solutions for any given fairness parameter α. These algorithms are obtained by adopting an alternating optimization based approach. The latter approach is well justified since the problem at hand is intractable and our goal is to obtain unified low-complexity algorithms that are suitable for all α.
- For the discrete user-association sub-problem, we first prove that this sub-problem itself is NP-hard and proceed to completely characterize the underlying set function that needs to be optimized. We then suggest and comprehensively analyze a simple centralized combinatorial algorithm (referred to as the GLS algorithm) that involves a Greedy stage followed by Local Search improvements. Our analysis yields meaningful and novel readily computable performance guarantees for all α. Previous related works have considered the proportional fairness (PF) utility and proposed combinatorial user association algorithms [12], [15]. Our results when specialized to the case of the weighted PF utility (by setting $\alpha = 1$) reveal that GLS is optimal up-to a constant additive factor of $-2\ln(2)$. Thus, a simple algorithm yields optimality up-to an additive constant factor, a fact that was hitherto only established for a significantly more complex algorithm [15] (whose run-time can depend on the input weights). Upon further specializing to the case with identical user weights, we see that the guarantee proved for a greedy algorithm in [12] has an instance dependent (non constant) additive factor. Interestingly, our simulation results indicate that in this special case the association yielded by GLS is identical to the optimal one obtained via another more complex algorithm from [12].
- We derive a distributed version of the GLS algorithm and prove that *remarkably* it provides guarantees identical to its centralized counterpart. This distributed version requires network assistance in the form of periodic broadcast of system load information similar to that proposed earlier in [27]. The main novelty of our approach is that we are able to configure each user to consider the system utility gain in contrast to the selfish gain used in the user-centric approach adopted by [18], [27] and more recently in [17]. Consequently, we can establish guarantees (with respect to the optimal system utility) and provable convergence for our distributed algorithms for all α. We note here that convergence of the user-centric approach to a Nash equilibrium was proved in [18] for particular choices of α and the recent and independent work in [17] has identified conditions under which the Nash equilibrium is (near-)optimal.
- For the continuous AF optimization sub-problem we adopt the auxiliary function method and show that it is provably convergent. Such a method has been used for precoder optimization originally over the single-cell downlink in [28] and over the multi-cell downlink in [7] followed by [6], [11]. We note however that unlike those works we incorporate fading coefficients that change at two different time scales. Further, a key step in our case entails a novel GP formulation, which we show can also be implemented in a distributed manner.

To improve readability the proofs of all the following propositions are deferred to [29].

II. Problem Statement

Considering the downlink in a HetNet, let $\mathcal{U} = \{1, \cdots, K\}$ denote the set of users and let $\mathcal{B}$ denote the set of transmission points (TPs) with cardinality $|\mathcal{B}| = B$. Further, suppose that the time axis is divided into multiple frames, where each frame consists of several consecutive slots. The fast fading coefficients for each user are assumed to change across slots in an independent identically distributed (i.i.d.) manner, while the slow fading coefficients are assumed to change across frames in an i.i.d. manner. *The choice of the activation fraction for each TP along with the user association for all TPs is made once for each frame to optimize the system utility*. This choice can be based on the slow fading realization in that frame but does not consider any previous such choices. Each TP then independently implements its per-slot scheduling policy over the users associated with it in that frame, where the latter scheduling policy respects the assigned activation fraction and can exploit the instantaneous fast fading coefficients seen by the associated users on each slot. Consequently, we can suppress the dependence on the frame and slot indices in the following.

In order to formulate an optimization problem for determining the user association and activation fractions, we derive an average rate that each user can obtain over a frame of interest, under any given user association and activation fractions. Towards this end, let $\mathcal{U}^{(b)},\ \forall b \in \mathcal{B}$ denote any given set of users associated to TP b over the frame and let $\boldsymbol{\rho} = [\rho_b]_{b\in\mathcal{B}}$ denote the activation vector, where $\rho_b \in [0,1]$ denotes the activation fraction assigned to TP b. We proceed by assuming that each TP b allocates a fraction $\gamma_{k,b} \in [0,1]$ of the frame to serve each associated user $k \in \mathcal{U}^{(b)}$, such that $\sum_{k\in\mathcal{U}^{(b)}} \gamma_{k,b} = 1$, where these fractions are determined at the onset of the frame. In particular, each TP is assumed to adopt an optimal fractional round robin per-slot scheduling policy. Note that an efficient per-slot scheduling policy (cf. [30]) that can adapt to the instantaneous fading and interference conditions seen across all the slots, will be at-least as good (in terms of optimizing the given utility). Next, we assume that the activation fraction of each TP b is implemented via a Bernoulli random variable $\mathcal{X}_b$ with $E[\mathcal{X}_b] = \rho_b$, that is i.i.d. across slots in the frame and is independent of all other random variables. Specifically, TP b is assumed to transmit (with a fixed power) when $\mathcal{X}_b = 1$ and remain silent otherwise. Then, an average rate that can be achieved for user $k \in \mathcal{U}^{(b)}$ is given by,

$$\gamma_{k,b}\rho_b \mathbb{E}\left[\log\left(1 + \frac{\beta_{k,b}}{1 + \sum_{b'\neq b}\beta_{k,b'}\mathcal{X}_{b'}}\right)\right] \qquad (1)$$

where the the desired channel gain $\beta_{k,b}$ and the interfering channel gains $\{\beta_{k,b'}\}$ are random variables that include both fast and slow fading as well as noise normalized transmit powers, and the expectation is over the activation variables

as well as the fast fading. Upon invoking the fact that the instantaneous rate in (1) is convex in the activation variables, which we recall are independent of the fast fading coefficients, we can further lower bound (1) to obtain

$$r_k = \gamma_{k,b}\,\underbrace{\rho_b \mathbb{E}\left[\log\left(1+\frac{\beta_{k,b}}{1+\sum_{b'\neq b}\beta_{k,b'}\rho_{b'}}\right)\right]}_{\triangleq R_{k,b}(\boldsymbol{\rho})}, \quad (2)$$

where now the expectation is over only the fast fading. Note that r_k in (2) depends on the slow fading realization (comprising of the path losses and shadowing factors) over the frame of interest. Letting $\mathbf{r} = [r_1, \cdots, r_K]$ denote the vector of such conservative rates obtained for all the K users over the frame, the achieved system utility is given by

$$\sum_{k\in\mathcal{U}} w_k u(r_k, \alpha), \quad (3)$$

where $\alpha > 0$ is a tunable fairness parameter and

$$u(r_k,\alpha) = \begin{cases} \frac{r_k^{(1-\alpha)}}{1-\alpha} & \alpha \in (0,1) \\ \log(r_k) & \alpha = 1 \\ -\frac{r_k^{(1-\alpha)}}{\alpha-1} & \alpha > 1 \end{cases} \quad (4)$$

and $w_k > 0$ denotes the weight of user $k \in \mathcal{U}$. These weights can be used to assign different priorities to different users and we assume that they are normalized, i.e., $\sum_{k\in\mathcal{U}} w_k = 1$. We can now write our problem, which is a mixed optimization problem, as

$$\boxed{\begin{aligned} &\max_{\substack{\boldsymbol{\rho}\in[0,1]^B; x_{k,b}\in\{0,1\};\\ \gamma_{k,b}\in[0,1]\ \forall\ k,b}} \left\{\sum_{k\in\mathcal{U}}\sum_{b\in\mathcal{B}} x_{k,b}\left(w_k u(\gamma_{k,b}R_{k,b}(\boldsymbol{\rho}))\right)\right\} \\ &\text{s.t.} \sum_{b\in\mathcal{B}} x_{k,b} = 1,\ \forall\ k\in\mathcal{U};\ \sum_{k\in\mathcal{U}}\gamma_{k,b} = 1\ \forall\ b\in\mathcal{B}. \end{aligned}} \quad (5)$$

Note that in (5) the binary variable $x_{k,b}$ is one if user k is associated to TP b and zero otherwise, so that the first set of constraints ensures that each user is associated with only one TP. Consequently, $\mathcal{U}^{(b)} \triangleq \{k : x_{k,b} = 1\}_{k\in\mathcal{U}}$ yields the user set associated with TP b. Note that in (5), we enforce $\{\mathcal{U}^{(b)}\}_{b\in\mathcal{B}}$ to be a partition of $\mathcal{U}$. This is meaningful and indeed important since we are targeting short-term optimality by maximizing a system utility independently over each frame. The joint optimization problem in (5) is unfortunately intractable. Consequently, we develop an alternating optimization framework to solve the joint problem in (5). We will demonstrate that although the user association and activation fractions are optimized assuming conservative rates and optimal fractional round robin per-slot scheduling policies at all TPs, the obtained solution retains its significant gains even without these assumptions.

III. User Association

We adopt the convention that $0\ln(0) = 0$ and consider any fixed activation vector $\boldsymbol{\rho}$ with strictly positive elements (otherwise any TP b with $\rho_b = 0$ can be simply removed). We proceed to systematically consider the user-association subproblem of (5) given by

$$\begin{aligned} &\max_{\substack{x_{k,b}\in\{0,1\};\\ \gamma_{k,b}\in[0,1]\ \forall\ k,b}} \left\{\sum_{k\in\mathcal{U}}\sum_{b\in\mathcal{B}} x_{k,b}\left(w_k u(\gamma_{k,b}R_{k,b}(\boldsymbol{\rho}))\right)\right\} \\ &\text{s.t.} \sum_{b\in\mathcal{B}} x_{k,b} = 1,\ \forall\ k\in\mathcal{U};\ \sum_{k\in\mathcal{U}}\gamma_{k,b} = 1\ \forall\ b\in\mathcal{B}, \end{aligned} \quad (6)$$

over three regimes defined by the values of α. We first define a ground set, $\underline{\Omega} = \{(k,b) : k \in \mathcal{U}, b \in \mathcal{B}\}$, that consists of all possible tuples and where each tuple (k,b) denotes an association of user k to TP b. Then, we also define the set $\underline{\Omega}^{(b)} = \{(k,b) : k \in \mathcal{U}\}$ for each TP $b \in \mathcal{B}$ which consists of all tuples whose TP is b, along with the set $\underline{\Omega}_{(k)} = \{(k,b) : b \in \mathcal{B}\}$ for each user k which consists of all tuples whose user is k. Finally, we define a family of sets $\underline{\boldsymbol{\mathcal{I}}}$, as the one which includes each subset of $\underline{\Omega}$ such that the tuples in that subset have mutually distinct users. Formally,

$$\underline{\mathcal{G}} \subseteq \underline{\Omega} : |\underline{\mathcal{G}} \cap \underline{\Omega}_{(k)}| \leq 1\ \forall\ k \Leftrightarrow \underline{\mathcal{G}} \in \underline{\boldsymbol{\mathcal{I}}}. \quad (7)$$

We start with the regime $\alpha > 1$ and note that for any given user association, i.e, for any given feasible choice of variables $\{x_{k,b}\}$, (6) is a continuous optimization problem. Moreover, it is separable across the set of TPs and for each TP $b \in \mathcal{B}$, we have a convex optimization problem over the set of variables $\{\gamma_{k,b}\}$ for $k \in \mathcal{U} : x_{k,b} = 1$. Using K.K.T. conditions it can be verified that for each TP $b \in \mathcal{B}$ [29]

$$\begin{aligned} &\max_{\substack{\gamma_{k,b}\in[0,1]\ \forall\ k\\ \sum_{k\in\mathcal{U}}\gamma_{k,b}=1}} \left\{\sum_{k\in\mathcal{U}} x_{k,b}\left(w_k u(\gamma_{k,b}R_{k,b}(\boldsymbol{\rho}))\right)\right\} = \\ &-\left(\sum_{k\in\mathcal{U}} x_{k,b}\left(w_k\frac{(R_{k,b}(\boldsymbol{\rho}))^{1-\alpha}}{\alpha-1}\right)^{1/\alpha}\right)^{\alpha} \end{aligned} \quad (8)$$

Consequently, upon defining

$$\Theta_k^{(b)}(\alpha) = \left(w_k\frac{(R_{k,b}(\boldsymbol{\rho}))^{1-\alpha}}{\alpha-1}\right)^{1/\alpha},\ \forall\ \alpha > 1,$$

(6) reduces to the following discrete optimization problem.

$$\min_{\substack{x_{k,b}\in\{0,1\}\ \forall\ k,b\\ \sum_{b\in\mathcal{B}} x_{k,b}=1\ \forall\ k}} \left\{\sum_{b\in\mathcal{B}}\left(\sum_{k\in\mathcal{U}} x_{k,b}\Theta_k^{(b)}(\alpha)\right)^{\alpha}\right\}. \quad (9)$$

Considering the case $\alpha \in (0,1)$, (6) reduces to

$$\max_{\substack{x_{k,b}\in\{0,1\}\ \forall\ k,b\\ \sum_{b\in\mathcal{B}} x_{k,b}=1\ \forall\ k}} \left\{\sum_{b\in\mathcal{B}}\left(\sum_{k\in\mathcal{U}} x_{k,b}\Theta_k^{(b)}(\alpha)\right)^{\alpha}\right\}, \quad (10)$$

where $\Theta_k^{(b)}(\alpha) = \left(w_k\frac{(R_{k,b}(\boldsymbol{\rho}))^{1-\alpha}}{1-\alpha}\right)^{1/\alpha},\ \forall\ \alpha \in (0,1)$.

Recalling the sets $\underline{\Omega}, \underline{\Omega}_{(k)}, \underline{\Omega}^{(b)}$ defined before, we further define the set function $g : 2^{\underline{\Omega}} \to I\!R$ as

$$g(\underline{\mathcal{G}},\alpha) = \sum_{b\in\mathcal{B}}\Big(\sum_{(k',b')\in\underline{\mathcal{G}}\cap\underline{\Omega}^{(b)}} \Theta_{k'}^{(b')}(\alpha)\Big)^{\alpha}, \quad (11)$$

$\forall\ \underline{\mathcal{G}} \subseteq \underline{\Omega}, \underline{\mathcal{G}} \neq \phi$ with $g(\phi,\alpha) = 0$, where ϕ denotes the empty set. The minimization problem in (9) is now re-formulated as

$$\min_{\underline{\mathcal{G}}:\underline{\mathcal{G}}\in\underline{\boldsymbol{\mathcal{I}}}\ \&\ |\underline{\mathcal{G}}|=K}\{g(\underline{\mathcal{G}},\alpha)\}, \quad (12)$$

whereas the maximization problem in (10) can be re-

formulated as

$$\max_{\underline{\mathcal{G}}:\underline{\mathcal{G}}\in\underline{\boldsymbol{\mathcal{I}}}\ \&\ |\underline{\mathcal{G}}|=K}\{g(\underline{\mathcal{G}},\alpha)\}. \quad (13)$$

Similarly, for $\alpha = 1$, (6) can be reformulated as in (13) but where $g(\phi,1)=0$ and for all $\underline{\mathcal{G}} \subseteq \underline{\Omega} : \underline{\mathcal{G}} \neq \phi$

$$\mathrm{g}(\underline{\mathcal{G}},1) = \sum_{(k,b)\in\underline{\mathcal{G}}} w_k \ln(w_k R_{k,b}(\boldsymbol{\rho})) - \sum_{b\in\mathcal{B}}\Big(\sum_{(k',b')\in\underline{\mathcal{G}}\cap\Omega^{(b)}} w_{k'}\Big)\ln\left(\sum_{(k',b')\in\underline{\mathcal{G}}\cap\Omega^{(b)}} w_{k'}\right). \quad (14)$$

We offer the following result.

Proposition 1: For any $\alpha > 0$, the user association sub-problem in (6) is NP-hard. Further, for any $\alpha > 1$, the set function $g(.,\alpha)$ is a normalized, non-negative and non-decreasing supermodular set function. For any $\alpha \in (0,1)$, the set function $g(.,\alpha)$ is a normalized, non-negative and non-decreasing submodular set function. The set function $g(.,1)$ is a normalized submodular set function.

Note that the set function $g(.,1)$ need not be non-negative nor non-decreasing.

A. GLS: A Unified Algorithm

In Table I we propose the GLS Algorithm, which is a simple combinatorial algorithm to solve the problem in (6). It considers the respective re-formulated versions in (12) or (13) and comprises of two stages. The first one is the greedy stage (steps 1 to 6). Here in each greedy iteration the feasible tuple (k',b') (with respect to the ones already selected so far) offering the best change in system utility is selected, until no such tuple can be found. In particular, (k',b') is determined as

$$\arg\max_{(k,b)\in\underline{\Omega}:\hat{\underline{\mathcal{G}}}\cup(k,b)\in\underline{\boldsymbol{\mathcal{I}}}}\{g(\hat{\underline{\mathcal{G}}}\cup(k,b),\alpha)-g(\hat{\underline{\mathcal{G}}},\alpha)\},\alpha\leq 1,$$
$$\arg\min_{(k,b)\in\underline{\Omega}:\hat{\underline{\mathcal{G}}}\cup(k,b)\in\underline{\boldsymbol{\mathcal{I}}}}\{g(\hat{\underline{\mathcal{G}}}\cup(k,b),\alpha)-g(\hat{\underline{\mathcal{G}}},\alpha)\},\alpha>1$$

The second stage of GLS is *local search improvement* and comprises of steps 7 to 13. Here, a feasible pair of tuples is determined in each local search iteration as $(k',b_1),(k',b_2) =$

$$\begin{cases}\arg\max\limits_{\substack{k\in\mathcal{U}\ \&\ b,b'\in\mathcal{B}\\(k,b)\in\breve{\underline{\mathcal{G}}},(k,b')\notin\breve{\underline{\mathcal{G}}}}}\{g(\breve{\underline{\mathcal{G}}}\cup(k,b')\setminus(k,b),\alpha)\},\alpha\leq 1,\\ \arg\min\limits_{\substack{k\in\mathcal{U}\ \&\ b,b'\in\mathcal{B}\\(k,b)\in\breve{\underline{\mathcal{G}}},(k,b')\notin\breve{\underline{\mathcal{G}}}}}\{g(\breve{\underline{\mathcal{G}}}\cup(k,b')\setminus(k,b),\alpha)\},\alpha>1\end{cases} \quad (15)$$

and the corresponding relative improvement is deemed to be better than Δ by checking if

$$g((\breve{\underline{\mathcal{G}}}\cup(k',b_2)\setminus(k',b_1)),\alpha)-g(\breve{\underline{\mathcal{G}}},\alpha) > \Delta\mathrm{sgn}(g(\breve{\underline{\mathcal{G}}},\alpha))g(\breve{\underline{\mathcal{G}}},\alpha),\ \alpha\leq 1, \quad (16)$$

$$g((\breve{\underline{\mathcal{G}}}\cup(k',b_2)\setminus(k',b_1)),\alpha)-g(\breve{\underline{\mathcal{G}}},\alpha) < -\Delta g(\breve{\underline{\mathcal{G}}},\alpha),\ \alpha>1, \quad (17)$$

where $\mathrm{sgn}(x)=1,\ \forall x\geq 0$ and -1 otherwise.

We now proceed to analyze the performance of GLS. We seek to bound the gap (by obtaining easily computable bounds) between the optimal system utility and the one returned by GLS. Towards this end, let $\underline{\mathcal{G}}^{\mathrm{opt}}$ denote the optimal solution to the problem in (12) for $\alpha > 1$ or (13) for $\alpha \in (0,1]$, and let

Table I: **GLS Algorithm**

1: Initialize with α, $\Delta \geq 0$, MaxIter ≥ 1, $\hat{\underline{\mathcal{G}}} = \phi$ and $\mathcal{U}' = \mathcal{U}$.
2: **Repeat**
3: Determine (k',b') as the tuple in $\underline{\Omega}$ which offers the best change among all tuples $(k,b)\in\underline{\Omega}$ such that $\hat{\underline{\mathcal{G}}}\cup(k,b)\in\underline{\boldsymbol{\mathcal{I}}}$.
4: Update $\hat{\underline{\mathcal{G}}} = \hat{\underline{\mathcal{G}}}\cup(k',b')$ and $\mathcal{U}' = \mathcal{U}'\setminus\{k'\}$
5: **Until** $\mathcal{U}' = \phi$.
6: Set $\breve{\underline{\mathcal{G}}} = \hat{\underline{\mathcal{G}}}$, Iter $= 0$.
7: **Repeat**
8: Increment Iter $=$ Iter $+ 1$.
9: Find a pair of tuples: $(k',b_1)\in\breve{\underline{\mathcal{G}}}$ and $(k',b_2)\in\underline{\Omega}\setminus\breve{\underline{\mathcal{G}}}$ such that the relative improvement upon swapping $(k',b_1)\in\breve{\underline{\mathcal{G}}}$ with (k',b_2) is better than Δ.
10: **If** such a pair exists then
11: Update $\breve{\underline{\mathcal{G}}} = \breve{\underline{\mathcal{G}}}\cup(k',b_2)\setminus(k',b_1)$.
12: **End If**
13: **Until** no such pair exists or Iter $=$ MaxIter.
14: Output $\breve{\underline{\mathcal{G}}}$.

$\breve{\underline{\mathcal{G}}},\hat{\underline{\mathcal{G}}}$ denote the counterparts obtained by our algorithm as the final output and after the greedy stage, respectively. We will first analyze the performance of the greedy first stage. The challenge here is that the underlying set function need not be submodular (when $\alpha > 1$) or it need not be non-negative and non-decreasing (when $\alpha = 1$), which makes the classical analysis in [31] inapplicable. To overcome this, we first derive new bounds that relate the optimal solution to that returned by the greedy stage. These bounds are in-fact applicable to arbitrary submodular or supermodular set functions. We then specialize those bounds to the set functions of interest to us in (11) and (14) to obtain the following result.

Proposition 2: For any given α, the greedy stage yields an output $\hat{\underline{\mathcal{G}}}$ such that

$$g(\hat{\underline{\mathcal{G}}},\alpha)\geq g(\underline{\mathcal{G}}^{\mathrm{opt}},\alpha)/2\ \ \forall\,\alpha\in(0,1),$$
$$g(\hat{\underline{\mathcal{G}}},1)\geq g(\underline{\mathcal{G}}^{\mathrm{opt}},1)-2\ln(2),$$
$$(3-2^{\alpha})g(\hat{\underline{\mathcal{G}}},\alpha)\leq g(\underline{\mathcal{G}}^{\mathrm{opt}},\alpha)\ \ \forall\,\alpha>1.$$

Remark 1: Note that the last bound in Proposition 2 is meaningful in the regime $\alpha\in\left(1,\frac{\ln(3)}{\ln(2)}\right)$ since then $3-2^{\alpha}>0$. As a result, we can deduce that for all $\alpha\in\left(0,\frac{\ln(3)}{\ln(2)}\right)$ the greedy stage of GLS itself provides firm (instance independent) guarantees. However, as α is increased, the performance of the greedy stage degrades compared to the optimal and the local search stage of GLS becomes increasingly important.

We now proceed to examine the performance of the local search stage. We leverage the techniques developed in [32] to analyze the behaviour of a local search based algorithm when the latter is used to maximize non-negative submodular functions. Here, we extend those techniques to arbitrary submodular and non-negative supemodular functions and also obtain sharper bounds. We let $\underline{e}=(k,b)$ denote any tuple in $\underline{\Omega}$ and expand $\breve{\underline{\mathcal{G}}}$ as $\breve{\underline{\mathcal{G}}}=\{\breve{\underline{e}}_1,\cdots,\breve{\underline{e}}_K\}$.

Proposition 3: The GLS algorithm for any given $\Delta\geq 0$ yields an output $\breve{\underline{\mathcal{G}}}$ such that for any given $\alpha>1$

$$g(\underline{\mathcal{G}}^{\mathrm{opt}},\alpha)\geq g(\breve{\underline{\mathcal{G}}},\alpha)+K(1-\Delta)g(\breve{\underline{\mathcal{G}}},\alpha)-h(\breve{\underline{\mathcal{G}}},\alpha)$$

and for any given $\alpha\in(0,1)$

$$g(\underline{\mathcal{G}}^{\mathrm{opt}},\alpha)\leq g(\breve{\underline{\mathcal{G}}},\alpha)+K(1+\Delta)g(\breve{\underline{\mathcal{G}}},\alpha)-h(\breve{\underline{\mathcal{G}}},\alpha).$$

Further, for $\alpha = 1$

$$g(\underline{\mathcal{G}}^{\rm opt}, 1) \leq$$
$$g(\underline{\breve{\mathcal{G}}}, 1) + K(1 + \Delta \text{sgn}(g(\underline{\breve{\mathcal{G}}}, 1)))g(\underline{\breve{\mathcal{G}}}, 1) - h(\underline{\breve{\mathcal{G}}}, 1).$$

where, $h(\underline{\breve{\mathcal{G}}}, \alpha) = \sum_{n=1}^{K} g(\underline{\breve{\mathcal{G}}} \setminus \underline{\breve{e}}_n, \alpha) + \sum_{n=1}^{K} (g(\tilde{\underline{\Omega}}, \alpha) - g(\tilde{\underline{\Omega}} \setminus \underline{\breve{e}}_n, \alpha))$, for any subset $\tilde{\underline{\Omega}} \subseteq \underline{\Omega} : \underline{\mathcal{G}}^{\rm opt} \cup \underline{\breve{\mathcal{G}}} \subseteq \tilde{\underline{\Omega}}$.
Finally, we note that one obvious choice of the subset $\tilde{\underline{\Omega}}$ needed in Proposition 3 is $\tilde{\underline{\Omega}} = \underline{\Omega}$. However, for $\alpha > 1$ this choice results in loose bounds and a better option is to set $\tilde{\underline{\Omega}}$ to be the set obtained after removing each tuple $\underline{e}$ satisfying $g(\underline{e}, \alpha) > g(\underline{\breve{\mathcal{G}}}, \alpha)$ from $\underline{\Omega}$. Note that no such tuple can be either in $\underline{\breve{\mathcal{G}}}$ or $\underline{\mathcal{G}}^{\rm opt}$. Note then that the bounds in Propositions 3 are easily computable once we have the output $\underline{\breve{\mathcal{G}}}$.

Regarding the complexity of GLS, it is easy to see that the complexity of the greedy stage is $O(K^2B)$. Moreover, each iteration in the local search (LS) stage has $O(BK)$ complexity. Further, simulation results presented later reveal that even for a large-sized HetNet ($KB \approx 3000$) only very few LS iterations (6 or less) are needed to capture the available gains.

B. Distributed Version

The GLS algorithm presented above assumes a centralized implementation. While this assumption is not very restrictive due to the fact that the implementation is done at a coarse time scale relying on average (not instantaneous) estimates, in practice a distributed implementation brings its own advantages. Remarkably, as we show next, for any given an activation vector $\boldsymbol{\rho}$, *a distributed variant of the GLS algorithm that offers identical performance guarantees is indeed possible.* We make a (justifiable) assumption that each user $k \in \mathcal{U}$ is supposed to know its weight w_k and its (single-user) rates $R_{k,b}(\boldsymbol{\rho}),\ \forall\ b \in \mathcal{B}$. Consequently, each user k can be configured to compute $\Theta_k^{(b)}(\alpha),\ \forall\ b \in \mathcal{B}$ given the fairness parameter α. $\Theta_k^{(b)}(\alpha),\ \forall\ k, b$ was defined before for all $\alpha \neq 1$ and here for later use we define $\Theta_k^{(b)}(1) = w_k,\ \forall\ k, b$. We will first derive a distributed version of the greedy stage of the GLS algorithm. Recall that in this stage a feasible subset of tuples $\underline{\hat{\mathcal{G}}}$ is built up. Then, we note the *simple but key fact* that given any subset of selected tuples $\underline{\hat{\mathcal{G}}} \in \underline{\mathcal{I}}$, the change in system utility upon adding a tuple $(k, b) \notin \underline{\hat{\mathcal{G}}}$ to $\underline{\hat{\mathcal{G}}}$, given by $g(\underline{\hat{\mathcal{G}}} \cup (k, b), \alpha) - g(\underline{\hat{\mathcal{G}}}, \alpha)$, can be expressed as

$$\begin{cases} \Theta_k^{(b)}(1) \ln(\Theta_k^{(b)}(1) R_{k,b}(\boldsymbol{\rho})) + \Psi^{(b)}(1) \ln(\Psi^{(b)}(1)) \\ \quad -(\Theta_k^{(b)}(1) + \Psi^{(b)}(1)) \ln(\Theta_k^{(b)}(1) + \Psi^{(b)}(1)), & \alpha = 1, \\ \quad (\Theta_k^{(b)}(\alpha) + \Psi^{(b)}(\alpha))^{\alpha} - (\Psi^{(b)}(\alpha))^{\alpha}, & \text{Else}, \end{cases}$$

where we define $\Psi^{(b)}(\alpha) = \sum_{(k',b') \in \underline{\hat{\mathcal{G}}} \cap \Omega^{(b)}} \Theta_{k'}^{(b')}(\alpha),\ \forall\ \alpha$. As a result, each user k (that has not associated to any TP yet) can compute the change in system utility if it joins any TP $b \in \mathcal{B}$, provided it knows $\Psi^{(b)}(\alpha)$, which we refer to as the current *load* on TP b. This suggests a natural distributed algorithm (outlined in Table II as the distributed greedy stage) comprising of two parts, namely, the TP-side and the user-side procedures. Considering the TP-side procedure, all TPs periodically broadcast their current load at the start of each time window on a designated slot, where the window size is chosen to accommodate all propagation, acknowledgement and processing delays, and where the broadcasting parameters (powers, assigned codes etc.) ensure that the loads can be reliably decoded by the users. We assume a particularly simple procedure where each TP admits only the first user (who has requested to associate) in each window. Moving to the user-side procedure, each user uses the current loads to determine the TP yielding the best system utility change, where the best change corresponds to the largest change for $\alpha \leq 1$ and to the smallest change for $\alpha > 1$. Note here that in each window (defined as the time interval between two consecutive load-broadcast slots) multiple associations can be done. Indeed, in each window, each TP that receives one or more user requests will admit one user, and each un-associated user will send one request. Hence, the distributed greedy stage will complete all associations in no more than K windows.

Table II: Distributed Greedy Stage

TP-side procedure: At each TP $b \in \mathcal{B}$
Repeat
Broadcast step:
  Transmit *current load* $\Psi^{(b)}(\alpha)$
Monitoring Step:
  If request from any user k detected
    If another user already admitted
      Send NACK to the requesting user k
    Else
      Admit user k and send an ACK
      Update current load $\Psi^{(b)}(\alpha) \rightarrow \Psi^{(b)}(\alpha) + \Theta_k^{(b)}(\alpha)$
    EndIf
  EndIf
Until No user request and no other TP changes its load
User-side procedure: At each user $k \in \mathcal{U}$
Repeat
Listening step:
  Decode all *current loads* $\Psi^{(b)}(\alpha),\ \forall\ b \in \mathcal{B}$
Request Step:
  Evaluate utility change upon joining each TP $b \in \mathcal{B}$
  Determine TP $\hat{b}$ corresponding to best change
  Send a request to associate to TP $\hat{b}$ along with $\Theta_k^{\hat{b}}(\alpha)$
Until ACK received from requested TP

Proposition 4: The solution obtained after the distributed greedy stage yields the same guarantees as in Proposition 2.
In [29] we also propose a distributed verision of the LS stage. We show that the distributed LS stage provably converges and the solution it yields upon convergence yields the same guarantees as in Proposition 3.

IV. AF OPTIMIZATION

The association scheme described in the previous section determines $\mathcal{U}^{(b)}$, the set of users associated to TP b for all $b \in \mathcal{B}$. In this section, for a given user association, we present a centralized algorithm to determine ρ_b for each b so as to optimize the system utility over different α regimes. For brevity we suppose that $\alpha > 1$. The analogous results for all other α values as well as an equivalent distributed variant of the proposed approach are deferred to [29]. The AF optimization problem in this regime is given by

$$\min_{\boldsymbol{\rho} \in [0,1]^B} \left\{ \sum_{b \in B} \left(\sum_{k \in \mathcal{U}^{(b)}} \tilde{w}_k / (R_{k,b}(\boldsymbol{\rho}))^{1-1/\alpha} \right)^{\alpha} \right\} \quad (18)$$

where $\tilde{w}_k = (\frac{w_k}{\alpha - 1})^{1/\alpha}$ and $R_{k,b}(\boldsymbol{\rho})$ is given by (2). We let $\boldsymbol{\beta}_k = \{\beta_{k,b}\}\ \forall b \in B$ denote the vector containing all

fading coefficients pertaining to user k on any slot. Then, we introduce auxiliary variables $g_{k,b}(\boldsymbol{\beta}_k)$ for each vector $\boldsymbol{\beta}_k$ for each user $k \in \mathcal{U}^{(b)}$ for each TP b. Using $g_{k,b}(\boldsymbol{\beta}_k)$ as a filter at user k to detect the signal transmitted from TP b over that slot, the mean squared error (MSE), $e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho})$, is given by

$$e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) = \left| g_{k,b}(\boldsymbol{\beta}_k)\sqrt{\beta_{k,b}} - 1 \right|^2 + |g_{k,b}(\boldsymbol{\beta}_k)|^2 + |g_{k,b}(\boldsymbol{\beta}_k)|^2 \sum_{b' \neq b} \beta_{k,b'}\rho_{b'} \tag{19}$$

Using the mutual information and MSE identity and introducing more auxiliary variables (cf. [28]), we have

$$R_{k,b}(\boldsymbol{\rho}) = \rho_b \mathbb{E}[\max_{g_{k,b}(\boldsymbol{\beta}_k), s_{k,b}(\boldsymbol{\beta}_k) \geq 0} \{1 - s_{k,b}(\boldsymbol{\beta}_k)e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) + \log(s_{k,b}(\boldsymbol{\beta}_k))\}] \tag{20}$$

The solution of each inner maximization problem in (20) is obtained by setting $g_{k,b}(\boldsymbol{\beta}_k)$ to be the MMSE filter $\hat{g}_{k,b}(\boldsymbol{\beta}_k)$ with $s_{k,b}(\boldsymbol{\beta}_k) = \hat{s}_{k,b}(\boldsymbol{\beta}_k) = 1/\hat{e}_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho})$, where $\hat{e}_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) = e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) \;|_{g_{k,b}(\boldsymbol{\beta}_k) = \hat{g}_{k,b}(\boldsymbol{\beta}_k)}$. Using (20), the problem in (18) (for the given association) can be re-formulated as the following optimization problem over variables $\boldsymbol{\rho}$, $\boldsymbol{s} = \{s_{k,b}(\boldsymbol{\beta}_k)\}, \boldsymbol{g} = \{g_{k,b}(\boldsymbol{\beta}_k)\}\ \forall \boldsymbol{\beta}_k, k \in \mathcal{U}^{(b)}, b \in B$.

$$\min_{\boldsymbol{\rho} \in [\mathbf{0},\mathbf{1}], \mathbf{g} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{1}} \left\{ \sum_{b \in B} \left(\sum_{k \in \mathcal{U}^{(b)}} \frac{\tilde{w}_k}{(\rho_b \mathbb{E}[1 - s_{k,b}(\boldsymbol{\beta}_k)e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) + \log(s_{k,b}(\boldsymbol{\beta}_k))])^{1-1/\alpha}} \right)^{\alpha} \right\} \tag{21}$$

Note that for a fixed $\boldsymbol{\rho}$, (21) can be optimized over $\mathbf{s}, \mathbf{g}$ via the closed form expressions given above. On the other hand, for fixed $\mathbf{s}, \mathbf{g}$ to optimize (21) over $\boldsymbol{\rho}$, we introduce additional variables $\boldsymbol{z} = \{z_b\}\ \forall b \in B$ and $\boldsymbol{t} = \{t_{k,b}\},\ \forall\ k \in \mathcal{U}^{(b)}, b \in B$ and express the reduced problem in (21) as

$$\min\nolimits_{\boldsymbol{\rho} \in [\mathbf{0},\mathbf{1}], \mathbf{z} \geq \mathbf{0}, \mathbf{t} \geq \mathbf{0}} \left\{ \sum_{b \in B} z_b^{\alpha} \right\}$$

subject to

$$z_b \geq \sum_{k \in \mathcal{U}^{(b)}} \tilde{w}_k t_{k,b}^{1/\alpha - 1} \quad \forall k, b$$

$$t_{k,b} \leq \rho_b \mathbb{E}[1 - s_{k,b}(\boldsymbol{\beta}_k)e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) + \log(s_{k,b}(\boldsymbol{\beta}_k))] \quad \forall k, b \tag{22}$$

Notice that (22) can in turn be re-written as

$$\min\nolimits_{\boldsymbol{\rho} \in [\mathbf{0},\mathbf{1}], \mathbf{z} \geq \mathbf{0}, \mathbf{t} \geq \mathbf{0}} \{ \sum_{b \in B} z_b^{\alpha} \}$$

subject to

$$\sum_k z_b^{-1} \tilde{w}_k t_{k,b}^{1/\alpha - 1} \leq 1 \quad \forall k, b \tag{23}$$

$$\frac{t_{k,b}\rho_b^{-1} + \mathbb{E}[s_{k,b}(\boldsymbol{\beta}_k)e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho})]}{1 + \mathbb{E}[\log(s_{k,b}(\boldsymbol{\beta}_k))]} \leq 1 \quad \forall k, b$$

The problem in (23) is a geometric program (GP) since all constraints are inequalities involving posynomials. Thus, we can repeat the following two steps until convergence.

1) Fix $\boldsymbol{\rho}$ and minimize (21) over $\boldsymbol{s}, \boldsymbol{g}$ using closed form solution of (20).
2) Fix $\boldsymbol{s}, \boldsymbol{g}$ and minimize (21) over $\boldsymbol{\rho}$ by solving GP in (23).

Note that since we have a monotonic improvement in the objective value of (21), convergence is guaranteed.

V. Joint Association & AF optimization

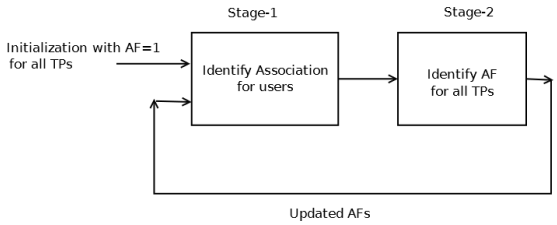


Figure 1: Joint Association and AF optimization block diagram

We propose two joint association & AF optimization algorithms for solving the problem in (5). These algorithms follow an alternating optimization approach where user association (stage-1) and AF (stage-2) are optimized in an alternating fashion. Fig. 1 shows a block-level decomposition. The first algorithm is the Joint GLS-AF algorithm, in which we first run the GLS algorithm (Algorithm in Table I) and use the obtained association in our AF optimization algorithm in Section IV. We repeat the following two steps until the benefit in terms of the alpha-fairness system utility falls below a threshold.

1) Stage1–Fix $\boldsymbol{\rho}$ and use GLS algorithm to calculate the user association.
2) Stage2–Fix the association and optimize over $\boldsymbol{\rho}$ using the auxiliary function method given in Section IV.

It is evident that both stages in the above alternating approach can be performed using the respective distributed versions that we derived before. However, one issue with the proposed joint GLS-AF algorithm, is that the TPs that do not serve any user in any one iteration will be discarded in all subsequent iterations. To overcome this potential limitation, we consider the joint relaxed association and AF (Joint RA-AF) algorithm. To obtain the association, this latter algorithm in stage-1 solves the convex optimization problem obtained by relaxing variables $x_{k,b},\ \forall\ k, b$ in (9) or (10) to be continuous variables in $[0, 1]$. In this solution, a user k can have $x_{k,b}$ non-zero for more than one TP b. In stage-2, the algorithm fixes $x_{k,b}$ for all k, b and optimizes the AF. To do so, it uses the auxiliary function method of Section IV on the objective in the problem (9) rather than (18) as $x_{k,b}$ can now have fractional values. This two stage procedure is repeated until the benefit in system utility falls below a threshold. Finally, the Joint RA-AF algorithm rounds $x_{k,b}$ to obtain a feasible association.

VI. Evaluation

We present a detailed evaluation of our proposed: Greedy Local Search (GLS) algorithm, the distributed Greedy (DG) algorithm and the joint association & AF optimization algorithms over an LTE HetNet deployment. In our evaluation topology an enhanced NodeB (eNB) covers the coordination area. The eNB site comprises of three cells (sectors), where each sector contains a set of eleven TPs formed by one macro and ten lower power (pico) nodes. We drop ninety nine users on the eNB site so there are a total of $B = 33$ TPs and

α	Greedy	GLS	RU	RRA	MSA	DG	LSI
0.25	67.75	67.82	67.82	67.82	65.08	67.48	1
0.5	112.67	112.67	112.71	112.52	107.03	110.39	0
0.75	288.57	288.57	288.82	288.46	277.65	283.98	0
1.0	-133.93	-133.87	-133.3 1	-133.93	-154.67	-139.76	1

Table III: Utility versus α

α	Greedy	GLS	LSI	α	Greedy	GLS	LSI
1.25	563.9	563.9	0	2.75	975.2	956.1	2
1.5	411.4	411.3	1	3.0	1345.8	1314.2	2
1.75	408.7	406.8	2	3.25	1904.6	1853.0	2
2.0	462.6	458.9	2	3.5	2754.6	2671.2	2
2.25	565.6	559.0	2	3.75	4045.1	3911.4	2
2.5	728.5	717.2	2	4.0	5953.6	5740.7	2

Table IV: Local Search Improvement

$K = 99$ users. All TPs and users have a single antenna each. We employ the conservative rates and ignore fast fading in the results presented in Section VI-A & Section VI-B. The results incorporating actual rates, fast fading and efficient per-slot user scheduling are presented later in Section VI-C.

A. Association

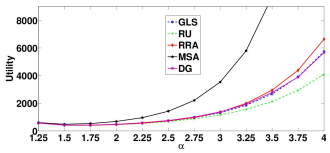
We compare the GLS & DG algorithms proposed in Section III-A and Section III-B, respectively, to the following:

- Relaxed Upperbound (RU)–Solves the convex optimization problem obtained by relaxing $x_{k,b}$ in (9) or (10). Though the obtained solution need not be feasible for (6), the scheme provides us with an upperbound on the optimal of (6).
- Relaxed Rounded Association (RRA)–Solves the convex optimization problem obtained by relaxing $x_{k,b}$ in (9) or (10). Each user k connects to the TP b corresponding to highest $x_{k,b}$ in the obtained convex optimization solution. This scheme is widely used to represent the performance that can be achieved by a feasible and near-optimal user association scheme. However, it requires solving a convex problem that can be computationally quite complex compared to GLS in a dense deployment.
- Max SNR Association (MSA)– Each user independently connects to the TP from which it sees the highest average channel gain. This scheme is the most common baseline.

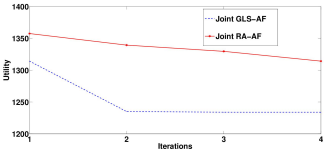
We evaluate the association algorithms by examining their returned utility function values for varying α. We also evaluate the additional gain yielded by the local search (LS) stage over the greedy one in the GLS algorithm.

1) $\alpha \leq 1$: We begin with an evaluation of GLS and the distributed greedy (DG) algorithm in the regime $\alpha \leq 1$, where we consider the maximization of the objective in (10). We set $\rho = 1$ for each of the 33 TPs and list the utility values of different association algorithms in Table III. As suggested by the guarantee in Proposition 2, we observe that greedy stage of GLS itself performs very close to the upper bound RU, and hence close to the optimal and provides good gains over the MSA scheme. Notice that GLS outperforms the RRA despite having a much lower computational complexity. Moreover, the DG algorithm performs close to the former two ones, while simultaneously offering the benefits of a distributed implementation. We also observe that the local search iterations (LSIs) of GLS are at-most 1 and that there is a slight utility gain obtained by the LS stage. Interestingly, upon employing the association algorithm from [12] we observed that the GLS indeed yields the optimal association for this example when $\alpha = 1$.

2) $\alpha > 1$: Next we study the performance of GLS & DG algorithms in $\alpha > 1$ region, where we consider the minimization of the objective in (9). As seen in Fig. 2(a) the proposed GLS & DG perform very similarly and they noticeably outperform RRA in $\alpha > 3$ regime while beating



2(a) Utility vs α



2(b) Utility vs iterations

MSA over the entire range of $\alpha > 1$. For example, GLS performs 13.5 % better than RRA and 80% better than MSA at $\alpha = 4$. MSA performs poorly throughout the $\alpha > 1$ regime since it has a naive user specific view rather than an optimized system specific view. The superiority of GLS & DG over RRA & MSA increases with increase in α. For example, at a high $\alpha = 10$, which approaches max-min fairness, the GLS outperforms RRA & MSA by 93.2% and 100% respectively. In Table IV we study the advantage of doing local search in the $\alpha > 1$ region. It is known that the greedy algorithm does not yield a constant factor approximation for the constrained minimization of a non-negative non-decreasing supermodular set function. [1] Therefore, the greedy stage need not be close to the optimal and there is room for improvement by the LS stage. As seen in Table IV, though the number of LS iterations are at-most 2, the order of gain over the greedy is upto 3.6%. At a higher $\alpha = 10$ the gain of GLS over greedy shoots up to 43%, with the number of LS iterations equal to 5. Therefore, as α is progressively increased, the local search stage of the GLS algorithm becomes increasingly important.

B. Joint Association & Activation fraction optimization

In Fig. 2(b) we study the performance of the two joint algorithms described in Section V for $\alpha = 3.0$ for up-to 4 iterations. Each point in the plot corresponds to an iteration, and is the utility value obtained using the updated association, where that association itself is calculated using the updated value of the activation fractions. The value at the first iteration is the utility corresponding to the association done using AF equal to 1 for all TPs. In the Joint RA-AF, at every iteration we calculate the utility by rounding the fractional association as done in the RRA algorithm. However, as mentioned in Section V, fractional values of the association variables $\{x_{k,b}\}$ are passed on to its second stage of AF identification. MSA with $\rho = 1$ for each TP with a utility value of 3531.8, performs much worse than the Joint GLS-AF & Joint RA-AF schemes. We obtain a gain of 6.1% for Joint GLS-AF over the case when we do only association via GLS with a fixed $\boldsymbol{\rho = 1}$, which demonstrates the benefit of doing the joint association and AF optimization. The Joint RA-AF scheme performs worse (upto 8.45%) than the Joint GLS-AF algorithm at every iteration,

[1]This problem is equivalent to the constrained maximization of a submodular set function albeit where that set function is not non-negative and non-decreasing, so that the classical result [31] is inapplicable.

illustrating that the benefits of GLS over RRA observed before at $\boldsymbol{\rho} = \mathbf{1}$ are preserved even in the joint optimization problem. For $\alpha = 0.5$, Joint GLS-AF performs 23.36% better than MSA with $\boldsymbol{\rho} = \mathbf{1}$, as compared to the gain of 4.6% obtained by GLS over MSA observed in Table III, again demonstrating the gain of optimizing AF and the association jointly. We observe that Joint GLS-AF & Joint RRA-AF algorithms perform very close to each other in $\alpha < 1$ regime. This is because of the similar performance of GLS and RRA schemes in this α regime.

C. Result Verification with Fast Fading

Finally, in this section we incorporate fast fading and efficient per-slot user scheduling to asses the benefits of the association and activation fractions calculated using proposed Joint GLS-AF algorithm. In particular, we assume that each frame comprises of 5000 slots and model all fast fading coefficients seen by each user on each slot as i.i.d. complex normal $\mathcal{CN}(0,1)$ variables. We randomly generate an ON-OFF pattern (for slots across each frame) for each TP that is compliant with its assigned activation fraction. Further, each TP employs the per-slot gradient based scheduling policy [30] over the set of users associated to it in order to maximize the utility. Then, using the actual per-user average rates so obtained, we compute the system utility values for different schemes. For $\alpha = 0.5$ we observed that the Joint GLS-AF scheme yields a 15.35% gain over the baseline scheme (MSA with $\boldsymbol{\rho} = \mathbf{1}$), while the gain of the GLS with $\boldsymbol{\rho} = \mathbf{1}$ over the baseline is 5.32%. For $\alpha = 3$ the gains of these two schemes over the baseline are 47.8% and 39.4%, respectively. This validates that our approach to obtain the association and AF does indeed result in significant gains in the presence of fast fading and efficient fine time-scale (per-slot) scheduling.

VII. Conclusion

We analyzed and evaluated novel association and activation fraction optimization algorithms for maximizing the alpha-fairness utility in HetNets. We derived useful performance guarantees and demonstrated the significant benefits of our proposed algorithms over a practical HetNet topology.

Appendix

We capture some basic definitions that are used in this paper.

Definition 1: Given a ground set Ω, we define its power set (i.e., the set containing all the subsets of Ω) as 2^{Ω}. Then, a real-valued function defined on the subsets of Ω, $h : 2^{\Omega} \rightarrow I\!R$ is normalized if $h(\phi) = 0$, where ϕ denotes the empty set. It is called a *submodular* set function if and only if

$$h(\mathcal{B} \cup a) - h(\mathcal{B}) \leq h(\mathcal{A} \cup a) - h(\mathcal{A}), \\ \forall\, \mathcal{A} \subseteq \mathcal{B} \subseteq \Omega \;\&\; a \in \Omega \setminus \mathcal{B}$$

and a *supermodular* set function if and only if

$$h(\mathcal{B} \cup a) - h(\mathcal{B}) \geq h(\mathcal{A} \cup a) - h(\mathcal{A}), \\ \forall\, \mathcal{A} \subseteq \mathcal{B} \subseteq \Omega \;\&\; a \in \Omega \setminus \mathcal{B}.$$

A non-negative valued set function $h : 2^{\Omega} \rightarrow I\!R_{+}$ is a non-decreasing set function when it satisfies, $0 \leq h(\mathcal{A}) \leq h(\mathcal{B}),\ \forall\, \mathcal{A} \subseteq \mathcal{B} \subseteq \Omega$.

References

[1] 3GPP, "Study on small cell enhancements for E-UTRA and E-UTRAN physical-layer aspects," *TR36.872 V12.0.0*, Sept. 2013.

[2] A. Gjendemsjoe, D. Gesbert, G. Oien, and S. Kiani, "Binary power control for sum rate maximization over multiple interfering links," *IEEE Trans. Wireless. Comm.*, Aug. 2008.

[3] W. Yu, T. Kwon, and C. Shin, "Multicell coordination via joint scheduling, beamforming and power spectrum adaptation," in *Proc. IEEE INFOCOM*, pp. 2570–2578, Apr. 2011.

[4] O. Ayach El, A. Lozano, and R. Heath, "On the overhead of interference alignment: Training, feedback, and cooperation," *IEEE Trans. on Wireless Comm.*, Nov. 2012.

[5] Y. Huang, G. Zheng, M. Bengtsson, K.-K. Wong, L. Yang, and B. Ottersten, "Distributed multicell beamforming with limited intercell coordination," *IEEE Trans. on Sig. Proc.*, Jan. 2011.

[6] M. Sanjabi, M. Razaviyayn, and Z. Q. Luo, "Optimal joint base station assignment and beamforming for heterogeneous networks," *IEEE Trans. on Sig. Proc.*, Apr. 2014.

[7] A. Tajer, N. Prasad, and X. Wang, "Robust linear precoder design for multi-cell downlink transmission," *IEEE Trans. Sig. Proc.*, jan 2011.

[8] N. Vaidhiyan, R. Subramanian, and R. Sundaresan, "Interference planning for multicell OFDM downlink," in *IEEE Comsnets (invited)*, 2011.

[9] H. Kim, G. de Veciana, X. Yang, and M. Venkatachalam, "Distributed $\alpha-$optimal user association and cell load balancing in wireless networks," *IEEE Trans. on Network.*, Feb. 2012.

[10] R. Madan, J. Borran, A. Sampath, N. Bhushan, A. Khandekar, and T. Ji, "Cell association and interference coordination in heterogeneous LTE-A cellular networks," *IEEE J. Sel. Areas Comm.*, Dec. 2010.

[11] K. Shen and W. Yu, "Distributed pricing-based user association for downlink heterogeneous cellular networks," *IEEE Journal Sel. Areas. Commun.*, Jun. 2014.

[12] N. Prasad, M. Arslan, and S. Rangarajan, "Exploiting cell dormancy and load balancing in LTE hetnets: Optimizing the proportional fairness utility," *IEEE Trans. on Commun.*, Oct. 2014.

[13] 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 Comm.*, June 2013.

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

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

[16] K. Son, S. Chong, and G. Veciana, "Dynamic association for load balancing and interference avoidance in multi-cell networks," *IEEE Trans. Wireless Comm.*, 2009.

[17] D. Bethanabhotla, O. Bursalioglu, H. Papadopoulos, and G. Caire, "Optimal user-cell association for massive mimo wireless networks," *v2, arXiv*, feb 2015.

[18] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, "RAT selection games in hetnets," in *IEEE Infocom*, 2013.

[19] N. Prasad, M. Arslan, and S. Rangarajan, "A two time scale approach for coordinated multi-point transmission and reception over practical backhaul," in *IEEE Comsnets (invited)*, jan 2014.

[20] R. Yates, "A framework for uplink power control in cellular radio systems," *IEEE JSAC*, Sep. 1995.

[21] E. Altman, A. Kumar, C. Singh, and R. Sundaresan, "Spatial sinr games of base station placement and mobile association," *IEEE Infocom.*, 2009.

[22] A. Bedekar and R. Agrawal, "Optimal muting and load balancing for eICIC," in *Proc. IEEE WiOPT*, 2013.

[23] S. Borst, S. Hanly, and P. Whiting, "Throughput utility optimization in hetnets," in *IEEE VTC*, 2013.

[24] I. Siomina and D. Yuan, "Analysis of cell load coupling for LTE network planning and optimization," *IEEE Trans. on Wireless Comm.*, June 2012.

[25] A. Fehske and G. Fettweis, "On flow level modeling of multi-cell wireless networks," in *IEEE WiOpt*, 2013.

[26] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *Proc. IEEE INFOCOM*, 2005.

[27] S. Deb, A. Keshavarz-Haddad, and V. Srinivasan, "MOTA: engineering an operator agnostic mobile service," in *IEEE Mobicom*, 2011.

[28] S. Christensen, R. Agarwal, E. Carvalho, and J. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 1–8, 2008.

[29] "Optimizing user association and activation fractions in heterogeneous wireless networks," *Tech. Rep: arXiv and http://bit.ly/1GAsfJt*, 2015.

[30] A. L. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multi-user throughput allocation," *Operations Res.*, 2005.

[31] G. L. Nemhauser and L. A. Wolsey, "Best algorithms for approximating the maximum of a submodular set function," *Math. Oper. Res.*, 1978.

[32] J. Lee, V. Mirrokni, V. Nagarajan, and M. Sviridenko, "Non-monotone submodular maximization under matroid and knapsack constraints," in *STOC*, 2009.