Distributed Base Station Association and Power Control for Heterogeneous Cellular Networks

Vu Nguyen Ha, *Student Member, IEEE*, and Long Bao Le, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a universal joint base station (BS) association (BSA) and power control (PC) algorithm for heterogeneous cellular networks. Specifically, the proposed algorithm iteratively updates the BSA solution and the transmit power of each user. Here, the new transmit power level is expressed as a function of the power in the previous iteration, and this function is called the power update function (puf). We prove the convergence of this algorithm when the puf of the PC strategy satisfies the so-called "two-sided scalable (2.s.s.) function" property. Then, we develop a novel hybrid PC (HPC) scheme by using noncooperative game theory and prove that its corresponding puf is 2.s.s. Therefore, this HPC scheme can be employed in the proposed joint BSA and PC algorithm. We then devise an adaptation mechanism for the HPC algorithm so that it can support the signal-to-interference-plus-noise ratio (SINR) requirements of all users whenever possible while exploiting multiuser diversity to improve the system throughput. We show that the proposed HPC adaptation algorithm outperforms the well-known Foschini–Miljianic PC algorithm in both feasible and infeasible systems. In addition, we present the application of the developed framework to design a hybrid access scheme for two-tier macrocell–femtocell networks. Numerical results are then presented to illustrate the convergence of the proposed algorithms and their superior performance, compared with existing algorithms in the literature.**

***Index Terms*—Femtocell networks, hybrid access, interference management, power control (PC), user association.**

I. Introduction

FEMTOCELLS have recently emerged as a potential solution to enhance indoor capacity and coverage [1]. Efficient deployment and operation of femtocells, however, demand to resolve various technical challenges beyond those existing in the traditional cellular network, including network architecture design, interference management, and synchronization issues. In fact, interference management for the macrocell–femtocell network is one of the most critical issues, which needs to be resolved to achieve enhanced network capacity and support users' quality of service (QoS) in different network tiers [1]–[5]. Since the emerging femtocells utilize the same licensed spectrum with existing macrocells, both cotier interference within each network tier (i.e., macro and femto tiers) and cross-tier interference among different network tiers need to be properly managed. For code-division multiple-access (CDMA) networks, power control (PC) and base station association (BSA) algorithms provide efficient mechanisms to mitigate interference and maximize the system throughput [10]–[13].

A good BSA algorithm typically provides an efficient mechanism to associate mobile users with one or several serving base stations (BSs) so that certain performance metrics of interest are optimized. In general, BSA can be decided based on different factors and metrics, such as achievable rates, transmit powers, geographical locations, and cell load [6]. In addition, the BSA can be jointly designed with PC [10]–[13]. In the two-tier macrocell–femtocell network, the design of a BSA algorithm also depends on the underlying access mode implemented at femtocells. Specifically, femtocells allow macro user equipment (MUE) devices to connect with their femto base stations (FBSs) in the open access, whereas they do not allow MUE devices to do so in the closed access [3], [4], [7]. In general, the open access is more effective in mitigating cross-tier interference compared with the closed access; however, it may result in uncontrollable performance degradation of femto users [3]. Design of an efficient hybrid access scheme that can balance between advantages and disadvantages of the other two access modes is, therefore, an interesting and important research topic.

PC is an important research topic that has been extensively investigated in the literature [8]–[22]. Among existing PC algorithms, there are two popular PC algorithms, namely, the target-signal-to-interference-plus-noise ratio (SINR)-tracking PC (TPC) algorithm [8] and the opportunistic PC (OPC) algorithm [14], [15]. Specifically, Foschini and Miljianic were the first to propose the distributed TPC algorithm that can support predetermined target SINRs for all users with minimum powers (i.e., achieving Pareto optimality) [8]. This PC algorithm was extended to consider maximum power constraints and distributed BS association in [9]–[12]. However, the TPC algorithm aims to support fixed target SINRs, which is, therefore, more applicable to the voice application. In contrast, the OPC algorithm, which was proposed by Sung and Leung in [14] and [15], aims at exploiting multiuser diversity to improve the system throughput. In these papers, Sung and Leung introduced the so-called "two-sided scalable (2.s.s.) function," which was the extended version of the "standard function" notion proposed by Yates [9], to prove the convergence of the OPC algorithm. Unfortunately, the OPC algorithm cannot provide any QoS support for users in the network. There have been also some more recent works that investigated the joint user removal and PC algorithms [16], [17]. In addition, various PC algorithms

Manuscript received April 12, 2012; revised July 30, 2012; November 19, 2012; February 15, 2013; and May 22, 2013; accepted July 10, 2013. Date of publication July 16, 2013; date of current version January 13, 2014. The review of this paper was coordinated by Dr. M. Dianati.

The authors are with the Institut National de la Recherche Scientifique—Énergie, Matériaux et Télécommunications, Université du Québec, Montréal, QC H3C 3P8, Canada (e-mail: hanguyen@emt.inrs.ca; long.le@emt.inrs.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2013.2273503

have been developed by employing the game theory approach [18]–[22]. In most cases, proposed PC algorithms are proved to converge to the Nash equilibrium (NE) of the underlying game. In particular, a pricing-based approach has been taken to design PC algorithms in [18] that strike to achieve an efficient throughput and a power tradeoff. In addition, sufficient conditions for convergence and having the largest number of users attain their target SINR have been presented.

There are some existing works that consider PC and interference management problems for femtocell networks [23]–[28]. In particular, Jo *et al.* [23] proposed two PC strategies, which aim to set the transmit powers of femto user equipment (FUE) devices to maintain the cross-tier interference below a fixed threshold at macrocell BSs (MBSs). Various PC strategies for both macrocells and femtocells were also developed by using the game theory approach in [24]–[26]. In particular, we previously proposed distributed PC algorithms for closed-accessed macro–femto networks in [26].

Interference management methods with dynamic subcarrier allocation and cell association were proposed in [27] and [28], respectively. Most existing works on interference management and PC for femtocell networks, however, assume the closed access [23]–[27], which are not flexible enough to support both voice and data users in different network tiers. This paper aims to resolve these limitations. Specifically, we make the following contributions.

- We develop a generalized BSA and PC algorithm and prove its convergence for any 2.s.s. power update function (puf). We then propose a hybrid PC (HPC) scheme that can be used in this general algorithm. In addition, we describe two alternative designs, namely, decomposed and joint BSA and PC design. For the decomposed design, we develop a simple load-aware BSA algorithm. These designs focus on the heterogeneous cellular network with different types of cells (e.g., macrocells, microcells, and femtocells) and users with different QoS requirements (e.g., voice and data users).
- We propose an HPC adaptation algorithm that adjusts parameters of the HPC scheme to support the differentiated SINR requirements of all users whenever possible while enhancing the system throughput. We then present the application of the proposed framework to the two-tier macrocell–femtocell networks.
- We present extensive numerical results to validate the developed theoretical results and to demonstrate the efficacy of our proposed algorithms.

The remainder of this paper is organized as follows: We describe the system model in Section II. In Section III, we present the generalized BSA and PC algorithm and the HPC scheme. We propose an HPC adaptation algorithm and present its application to macrocell–femtocell networks in Section IV. Numerical results are presented in Section V, followed by the conclusion in Section VI.

II. System Model

We consider uplink communications in a heterogeneous wireless cellular network. We assume that there are M users, which are labeled $1, 2, \ldots, M$, transmitting information to K BSs, which are labeled $1, 2, \ldots, K$, on the same spectrum by using CDMA. Each of these BSs can belong to one of the available cell types (e.g., macrocells, microcells, picocells, and femtocells) in a heterogeneous cellular network. Let $\mathcal{M}$ and $\mathcal{K}$ be the sets of all users and BSs, respectively, i.e., $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$. Assume that each user i communicates with only one BS at any time, which is denoted by $b_i \in \mathcal{K}$. However, users can change their associated BSs over time. Note that if $b_i \equiv b_j$, then users i and j are associated with same BS. Let D_i be the set of BSs that user i can be associated with, which are nearby BSs of user i in practice. Note that we have $D_i \subseteq \mathcal{K}$ and $b_i \in D_i$. We are interested in developing a BSA strategy that determines how each user i will choose one BS $b_i \in D_i$ to communicate with based on its observed channel state information and interference.

Let the transmit power of user i be p_i, whose maximum value is $\bar{p}_i$, i.e., $0 \leq p_i \leq \bar{p}_i$. We arrange transmit powers of all users in a vector, which is denoted by $p = (p_1, p_2, \ldots, p_M)$. Let h_{ij} be the channel power gain from user j to BS i and η_i be the noise power at BS i. Then, the SINR of user i at BS b_i can be written as [29]

$$\Gamma_i(p) = \frac{G h_{b_i i} p_i}{\sum_{j \neq i} h_{b_i j} p_j + \eta_{b_i}} = \frac{p_i}{R_i(p, b_i)} \tag{1}$$

where G is the processing gain, which is defined as the ratio of the spreading bandwidth to the symbol rate; $R_i(p, b_i)$ is the effective interference to user i, which is defined as

$$R_i(p, b_i) \triangleq \frac{\sum_{j \neq i} h_{b_i j} p_j + \eta_{b_i}}{G h_{b_i i}} \tag{2}$$

where $R_i(p, k)$ is the effective interference experienced by user i at BS k. We will sometimes write $R_i(p)$ instead of $R_i(p, b_i)$ when there is no confusion. We assume that each user i requires the minimum QoS in terms of a target SINR $\hat{\gamma}_i \ \forall\, i \in \mathcal{M}$. In particular, these QoS requirements can be written as follows:

$$\Gamma_i(p) \geq \hat{\gamma}_i, \quad i \in \mathcal{M}. \tag{3}$$

The objective of this paper is to develop distributed BSA and PC algorithms that can maintain the SINR requirements in (3) (whenever possible) while exploiting the multiuser diversity gain to increase the system throughput. The proposed algorithms, therefore, aim to support both voice and high-speed data applications. Moreover, we aim to achieve these design objectives for a heterogeneous wireless environment where there are different kinds of users with differentiated QoS targets (e.g., voice and data users) and potentially different cell types (e.g., macrocells, microcells, picocells, and femtocells). In particular, voice users would typically require some fixed target SINR $\hat{\gamma}_i$, whereas data users would seek to achieve higher target SINR $\hat{\gamma}_i$ to support their broadband applications (e.g., video and Internet browsing).

III. Base Station Association and Power Control

A. Generalized BSA and PC Algorithm

We develop a general minimum effective interference BSA and PC algorithm and prove its convergence. Specifically, we will focus on a general iterative PC algorithm where each user i in the network performs the following power update $p_i^{(n+1)} := J_i(p^{(n)})$, where n denotes the iteration index, and $J_i(.)$ is the puf.[1] In fact, this kind of PC algorithm converges if we can prove that its corresponding puf is a *2.s.s. function* according to Sung and Leung [14], [15]. The challenges involved in designing such a PC algorithm are that we have to ensure its puf is *2.s.s.*, it fulfills our design objectives for the heterogeneous cellular network, and it can be implemented in a distributed manner. In addition, we seek to design a PC algorithm that can be integrated with an efficient BSA mechanism. Toward this end, we give the definition of the 2.s.s. function in the following.

Definition 1: A puf $J(p) = [J_1(p), \ldots, J_M(p)]^T$ is 2.s.s. with respect to p if for all $a > 1$ and any power vector p' satisfying $(1/a)p \leq p' \leq ap$, we have

$$(1/a)J_i(p) < J_i(p') < aJ_i(p) \quad \forall\, i \in \mathcal{M}. \tag{4}$$

We will consider a joint BSA and PC algorithm where the puf of the PC scheme satisfies the 2.s.s. property stated in Definition 1. Under this design, each user chooses its "best" BS and updates its transmit power in a distributed manner. Specifically, the proposed algorithm is described in Algorithm 1 where each user chooses a BS that results in minimum effective interference and updates its power by using any 2.s.s. puf $J(p) = J'(R(p)) = [J'_1(R_1(p)), \ldots, J'_M(R_M(p))]$, where $R(p) = [R_1(p), \ldots, R_M(p)]$. The proposed algorithm can be distributively implemented with any distributed PC algorithm. To realize the BSA, each user needs to estimate the effective interference levels for different nearby BSs of interest. Then, each user i can choose one BS k in D_i with a minimum value of $R_i(p, k)$. This algorithm ensures that each user experiences low effective interference and, therefore, high throughput at convergence.

User i can estimate $R_i(p, k)$ if it has information about the total interference and noise power [i.e., the denominator of (1)] and the channel power gain h_{ki} according to the SINR expression in (1) (since the current transmit power level p_i is readily available). In addition, the channel power gains h_{ki} can be estimated by BS k and sent back to each user by using the pilot signal and any standard channel estimation technique [30]–[32]. Moreover, the total interference and noise power for each user can be estimated as follows. Each BS estimates the total received power and then broadcasts this value to its connected users. Each user i can calculate the total interference and noise power by subtracting its received signal power (i.e., $h_{ki}p_i$) from the total receiving power broadcast by the BS. Therefore, calculation of the effective interference only requires the standard channel estimation of h_{ki} and estimation of the total receiving power at the BS. In addition, signaling is only involved in sending these values from the BS to its connected users, which is relatively mild and can be conducted over the air. More importantly, estimation of the effective interference $R_i(p)$ and, therefore, the proposed PC algorithm given in (13) can be implemented in a distributed fashion.

[1] It can be verified that the pufs of the TPC and OPC schemes are 2.s.s.

Algorithm 1 Minimum Effective Interference BS Association and PC Algorithm

---

1: Initialization:
- $p_i^{(0)} = 0$ for all user i, $i \in \mathcal{M}$.
- $b_i^{(0)}$ is set as the nearest BS of user i.

2: Iteration n: Each user $i (i \in \mathcal{M})$ performs the following:
- Calculate the effective interference at BS $k \in D_i$ as follows:

$$R_i^{(n)}\left(p^{(n-1)}, k\right) = \frac{\sum_{j \neq i} h_{kj} p_j^{(n-1)} + \eta_k}{G h_{ki}}. \tag{5}$$

- Choose the BS $b_i^{(n)}$ with the minimum effective interference, i.e.,

$$b_i^{(n)} = \operatorname{argmin}_{k \in D_i} R_i^{(n)}\left(p^{(n-1)}, k\right) \tag{6}$$

$$R_i^{o(n)}\left(p^{(n-1)}\right) = \min_{k \in D_i} R_i^{(n)}\left(p^{(n-1)}, k\right) = R_i^{(n)}\left(p^{(n-1)}, b_i^{(n)}\right). \tag{7}$$

- Update the transmit power for the chosen BS as follows:

$$p_i^{(n)} = J'_i\left(R_i^{o(n)}\left(p^{(n-1)}\right)\right) \tag{8}$$

where $J'_i(R_i(p))$ is the puf with respect to $R_i(p)$.

3: Increase n and go back to step 2 until convergence.

Note that we have expressed the pufs with respect to both p and $R(p)$. For the expression with respect to $R(p)$, the ith element of the puf is denoted by $J'_i(R_i(p))$, where $R_i(p)$ is the effective interference experienced by user i given in (2). Therefore, we have $J(p) = J'(R(p))$, which depends on both the transmit powers of all users and the BS with which each user i is associated in general. We establish the convergence of Algorithm 1 in the following by utilizing the 2.s.s. function approach. Toward this end, we recall the convergence result for any PC algorithm that employs a bounded 2.s.s. puf in the following lemma [14].

Lemma 1: Assume that $J(p)$ is a 2.s.s. function, whose elements $J_i(p)$ are bounded by zero and $\bar{p}_i$, i.e., $0 \leq J_i(p) \leq \bar{p}_i$. Consider the corresponding power update $p_i^{(n+1)} := J_i(p^{(n)})$, $\forall\, i$, where n denotes the iteration index. Then, we have the following results.

1) *The puf $J(p)$ has a unique fixed point corresponding to a transmit power vector p^* that satisfies $p^* = J(p^*)$.*

2) *Given an arbitrary initial power vector* $p^{(0)}$, *the PC algorithm based on puf* $J(p)$ *converges to that unique fixed point* p^*.

Proof: The results stated in this lemma have been established for the 2.s.s. puf in [14]. ■

We are now ready to state one important result for the joint BSA and PC operations described in Algorithm 1 in the following theorem.

Theorem 1: Assume that $J(p)$ and $J'(R(p))$ are arbitrary 2.s.s. pufs with respect to p and $R(p)$, respectively. Then, Algorithm 1 converges to an equilibrium.

Proof: The proof is given in Appendix A. ■

To prove that Algorithm 1 converges, we first show that the puf $J^{\circ}(p) = [J_1^{\circ}(p), \ldots, J_M^{\circ}(p)]$ is a 2.s.s. function with respect to p, where $J_i^{\circ}(p) = J_i'(R_i^{\circ}(p))$, and $R_i^{\circ}(p) = \min_{k \in D_i} R_i(p,k)$, $\forall\, i \in \mathcal{M}$. Then, the convergence can be established by applying the results in Lemma 1. The result in this theorem implies that if pufs $J(p)$ and $J'(R(p))$ are 2.s.s., then the BSA strategy described in (6) and (7) results in a composite 2.s.s. puf, which corresponds to the joint BSA and PC operation. In other words, the proposed BSA scheme proposed in Algorithm 1 preserves the 2.s.s. property of the employed puf $J(p)$.

B. HPC Algorithm

To complete the design presented in Algorithm 1, we need to develop a distributed PC strategy, which is employed in (8). In general, the performance of a PC algorithm depends on how we design the corresponding 2.s.s. puf $J(p)$. We will propose a new puf, which is denoted by $I^H(p)$, and the corresponding PC algorithm in the following.

1) Game-Theoretic Formulation: We develop the distributed PC algorithm by using the noncooperative game theory approach. In particular, we define a PC game as follows.

- Players: This is the set of mobile users $\mathcal{M}$.
- Strategies: Each user i chooses transmit power in set $[0, \bar{p}_i]$.
- Payoffs: User i is interested in maximizing the following payoff function:

$$U_i(p) \triangleq -\alpha_i \left(p_i - \xi_i R_i(p)^{\frac{x}{x-1}}\right)^2 - \left(p_i - \widehat{\gamma}_i R_i(p)\right)^2 \quad (9)$$

where $\widehat{\gamma}_i$ denotes the target SINR for user i, x is a special parameter whose desirable value will be revealed in Theorem 2, and α_i and ξ_i are nonnegative control parameters, i.e., $\alpha_i, \xi_i \geq 0$, which will be adaptively adjusted to achieve our design objectives.

This game-theoretic formulation arises quite naturally in autonomous spectrum access scenarios, such as heterogeneous wireless networks where mobile users tend to be selfish and are only interested in maximizing their own benefits. Using this formulation, we will develop an iterative PC algorithm in which each user maximizes its own payoff in each iteration given the chosen power levels from other users in the previous iteration (i.e., each user plays the *best response* strategy). To devise such an algorithm, each user i chooses the power level, which is obtained by setting the first derivative of the underlying user's payoff function to zero.

In fact, by maximizing the payoff function given in (9), each user i strikes to balance between achieving the SINR target $\widehat{\gamma}_i$ and exploiting its potential favorable channel condition to increase its SINR. While it is quite intuitive that maximizing $-(p_i - \widehat{\gamma}_i R_i(p))^2$ enables user i to reach its target SINR $\widehat{\gamma}_i$, the design intuition in optimizing the first term $-\alpha_i(p_i - \xi_i R_i(p)^{x/x-1})^2$ may not be very straightforward. We state one result, which reveals the engineering intuition behind this in the following lemma.

Lemma 2: Consider the game formulation previously described with infinite power budget (i.e., $\bar{p}_i = \infty\, \forall\, i$). Then, best responses due to the following two payoff functions are the same:

$$U_i^{(1)}(p) \triangleq \Gamma_i^x - \lambda_i p_i \quad (10)$$

$$U_i^{(2)}(p) \triangleq -\left(p_i - \xi_i R_i(p)^{\frac{x}{x-1}}\right)^2 \quad (11)$$

if $\xi_i = (\lambda_i/x)^{1/x-1}$ and $0 < x < 1$, where λ_i represents the pricing coefficient of user i.

Proof: The proof is given in Appendix B. ■

Remark 1: It has been shown that the OPC algorithm can be achieved if users iteratively play their corresponding best response strategies with payoff function $U_i^{(1)}(p)$ for $x = 1/2$ [14], [15]. Moreover, if each user i plays the best response strategies using $U_i(p)$ with $\alpha_i = 0$, then we can obtain the well-known TPC algorithm. Therefore, our chosen payoff function in (9) can be used to design an HPC strategy that exploits the advantages of both OPC and TPC algorithms.

2) Proposed HPC Algorithm: We are now ready to develop an HPC algorithm corresponding to the payoff function in (9). Specifically, we can derive the power update rule for the HPC algorithm according to the best response strategy of the underlying payoff function. After some manipulations, we can obtain the following best response under the chosen payoff function (9):

$$p_i = I_i(p) \triangleq \frac{\alpha_i \xi_i R_i(p)^{\frac{x}{x-1}} + \hat{\gamma}_i R_i(p)}{\alpha_i + 1}. \quad (12)$$

Considering the maximum power constraints, the HPC algorithm employs the following iterative power update:

$$p_i^{(n+1)} = I_i^H\left(p^{(n)}\right) = \min\left\{\bar{p}_i, I_i(p^{(n)})\right\} \quad (13)$$

where n denotes the iteration index, and $I_i(p)$ is given in (12). Here, parameter α_i can be used to control the desirable performance of the proposed HPC algorithm. Specifically, by setting $\alpha_i = 0$, user i actually employs the standard Foschini–Milijanic TPC algorithm to achieve its target SINR $\hat{\gamma}_i$, whereas if $\alpha_i \to \infty$, user i attempts to achieve a higher SINR (if it is in favorable condition). It can be observed that each user i only needs to calculate or estimate the effective interference $R_i(p)$ to update its power in the proposed HPC algorithm.

3) Convergence of HPC Algorithm: Here, we establish the convergence condition for the proposed HPC algorithm by using the *2.s.s. function* approach given in Definition 1. We state a sufficient condition under which puf $I^H(p)$ in (13) is 2.s.s., as in Theorem 2.

Theorem 2: If parameter x of function $I_i(p)$ given in (12) satisfies $0 < x \leq 1/2$, then puf $I_i^H(p)$ given in (13) is 2.s.s. In addition, the proposed HPC algorithm in (13) converges to the NE of the underlying PC game.

Proof: The convergence of the proposed HPC algorithm immediately follows from the results of Lemma 1 if we can prove that its puf $I^H(p) = [I_1^H(p), \ldots, I_M^H(p)]$ in (13) is 2.s.s. with respect to p. In addition, the resulting equilibrium (i.e., the power vector at convergence) is the NE of the PC game defined in Section III-B1 since users play the best response strategy. We will prove that $I^H(p)$ is 2.s.s. in Appendix C. ■

The proposed HPC scheme will be employed in Algorithms 1 and 3 presented in this paper. Note, however, that this proposed HPC scheme can be used as a standalone PC algorithm in general.

C. Alternative BSA and PC Designs

We describe two alternative designs for the BSA and PC operations in the following.

1) Decomposed BSA and PC Design: We can design the BSA and PC algorithms separately and implement them over different time scales. In particular, a BSA solution can be obtained and fixed based on the average long-term channel state information. Then, PC is applied to the BSA solution to achieve the design objectives. The advantage of this decomposed design is that BSA is only updated as the long-term channel state information significantly changes. Therefore, this design requires infrequent updates of the BSA solution. This would lead to reduced computation and signaling complexity.

Algorithm 2 LOAD-AWARE BSA ALGORITHM

1: Each BS estimates the average channel power gains from nearby users to itself.
2: By assuming all users transmit with their maximum powers, each BS estimates/calculates SINRs achieved by nearby users; it transmits these estimated SINRs to them.
3: Upon receiving estimated SINRs from all potential BSs, each user will associate with the BS achieving the largest estimated SINR.

We propose a load-aware BSA algorithm for this design in Algorithm 2. In this algorithm, each user is associated with the BS that results in the highest SINR by using the average channel power gains and assuming maximum transmit powers. To implement this algorithm, each BS needs to estimate the SINRs for nearby users and broadcasts these estimated SINRs to them, and based on this, each user chooses the associated BS. Given the BSA solution, users run the HPC strategy to settle the power levels. This decomposed design would be more applicable to the fast-fading environment where the joint dynamic BSA and PC algorithm may not work efficiently.

2) Joint BSA and PC Design: In general, the network throughput can be improved by performing the joint BSA and PC algorithm presented in Algorithm 1. In particular, the BSA solution is dynamically updated jointly with PC, which employs the proposed HPC scheme presented in Section III-B. Here, each user i chooses one BS that achieves the minimum effective interference $R_i(p,k)$ and updates its power level accordingly. When a particular user is in the common neighborhood of several BSs, the fluctuation of its channel power gains toward these BSs may result in varying BSA decisions even for fixed users. This means that each user may transmit data to different BSs over a short time interval. In addition, to facilitate the dynamic BSA, each user needs to estimate the effective interference frequently. This requires more signaling overhead in the control channel compared with the decomposed BSA and PC design.

IV. HYBRID POWER CONTROL ADAPTATION ALGORITHM

The equilibrium achieved by the proposed HPC scheme at convergence for either decomposed or joint BSA and PC design depends on its parameters, namely, α_i and ξ_i for $i \in \mathcal{M}$. Here, we develop decentralized mechanisms to adjust these parameters so that target SINRs of all users can be achieved whenever possible while enhancing the system throughput. The proposed adaptive mechanisms comprise two updating operations in two different time scales, i.e., running HPC algorithm (and the appropriate BSA strategy) to achieve the NE point in the small time scale and updating α_i and ξ_i for all users to achieve desirable NE in the large time scale. Moreover, we discuss the application of the proposed framework to the two-tier macrocell–femtocell networks. Let $\Delta = \{\alpha_i \mid i \in \mathcal{M}\}$ and $\Xi = \{\xi_i \mid i \in \mathcal{M}\}$ be the set of α and ξ parameters of all users in the puf of the HPC scheme, respectively.

A. Two Time-Scale Adaptive Algorithm

As discussed in Remark 1, the TPC scheme is a special case of the proposed HPC scheme with $\alpha_i = 0$ for $i \in \mathcal{M}$. It is known that the TPC scheme is able to support all users' target SINRs expressed in (3), as long as the system is feasible [i.e., the SINR requirements in (3) can be supported] [16]. However, the TPC scheme fails to achieve high system throughput when the system is feasible and lowly loaded. In addition, the TPC scheme may not be able to support the largest possible number of users when the system is infeasible. Our objective is to develop an adaptive strategy for the proposed HPC algorithm to overcome these limitations of the TPC scheme.

Toward this end, if user i is a voice user who is only interested in maintaining its target SINR $\hat{\gamma}_i$, then we can simply set $\alpha_i = 0$ in (9). For each data user i, we will fix ξ_i while adaptively updating power and α_i in two different time scales to achieve the design objectives. Specifically, each data user i will run the HPC and BSA algorithm for a given α_i until convergence (i.e., in the small time scale); then, it updates α_i accordingly (i.e., in the large time scale). To set the value for ξ_i, suppose that data user i would need to use its maximum power

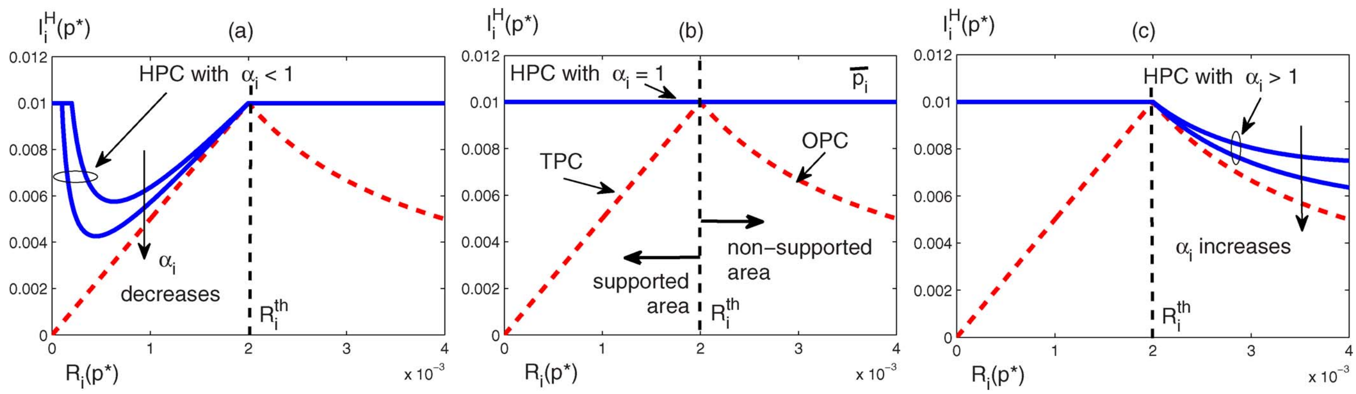

Fig. 1. Relationship between HPC puf $I_i'^H(R_i(p))$ and $R_i(p)$ for different values of α_i ($\bar{p}_i = 0.01$ W), where $R_i(p^*) \le R_i^{\text{th}}$ for supported users and $R_i(p^*) > R_i^{\text{th}}$ for nonsupported users.

$\bar{p}_i$ to reach the target SINR $\hat{\gamma}_i$. Then, the value of ξ_i can be found using the result in (26), shown in Appendix B, as

$$\xi_i = (\bar{p}_i/\hat{\gamma}_i^x)^{\frac{1}{1-x}} \tag{14}$$

where we have substituted $\Gamma_i = \hat{\gamma}_i$ to (26). We now describe how to update α_i in Δ. In the HPC scheme, puf $I_i'^H(R_i(p)) = I_i^H(p)$ depends on the effective interference $R_i(p)$ and the value of α_i. Let p^* denote the power vector at convergence, which is obtained by running the proposed HPC algorithm for a particular vector α. We illustrate the relationship between $I_i'^H(R_i(p^*))$ and $R_i(p^*)$ in Fig. 1, where $R_i^{\text{th}} = \bar{p}_i/\hat{\gamma}_i$ is a threshold effective interference of user i. We reveal the relationship between $\Gamma_i(p^*)$, $R_i(p^*)$, and α_i in the following lemma.

Lemma 3: Assume that $\xi_i = (\bar{p}_i/\hat{\gamma}_i^x)^{1/1-x}$ and $0 < x \le 1/2$. Then, we have the following.

1) If $R_i(p^*) > R_i^{\text{th}}$, then $\Gamma_i(p^*) < \hat{\gamma}_i, \forall\, \alpha_i \ge 0$.
2) If $R_i(p^*) \le R_i^{\text{th}}$, then $\Gamma_i(p^*) \ge \hat{\gamma}_i, \forall\, \alpha_i \ge 0$.
3) If $R_i(p^*) < R_i^{\text{th}}$, then we have
   - $\Gamma_i(p^*) = \hat{\gamma}_i$ iff $\alpha_i = 0$.
   - $I_i(p^*)$ decreases if α_i decreases.

Proof: The proof is given in Appendix D. ■

It is shown in Fig. 1 that if $\alpha_i < 1$, HPC curves become closer to the TPC curve as α_i decreases, whereas if $\alpha_i > 1$, HPC curves become closer to the TPC curve as α_i increases. This is because $I_i(p)$ in the puf of the proposed HPC scheme is a weighted sum of those in the TPC and OPC schemes. Recall that our design objectives are to maintain the SINR requirements for all users expressed in (3) (whenever possible) while enhancing the system throughput. Let user i be a supported user (nonsupported user) if its SINR is greater (less) than its target SINR, which occurs as its effective interference $R_i(p^*)$ is less (greater) than threshold R_i^{th}, respectively. Note that a supported user i will have its SINR greater than the target SINR if $\alpha_i > 0$. We refer to such a supported user as a potential user in the following. In fact, any potential user i can reduce its transmit power to enhance the SINRs of other users (since this reduces the effective interference experienced by other users) as being implied by the results in Lemma 3. Therefore, by adjusting α_i, each user i can vary its SINR and assist other users in improving their SINRs.

Algorithm 3 HPC ADAPTATION ALGORITHM

1: Initialization:
- Set $p^{(0)} = 0$, i.e., $p_i^{(0)} = 0, \forall\, i \in \mathcal{M}$.
- Set $\Delta^{(0)}$ as $\alpha_i^{(0)} = 0$ for voice user and $\alpha_i^{(0)} = \alpha_0$ ($\alpha_0 \gg 1$) for data user.
- Set $\overline{N}^* = |\overline{\mathcal{M}}^{(0)}|$ and $\Delta^* = \Delta^{(0)}$.

2: Iteration l:
- Run HPC algorithm until convergence with $\Delta^{(l)}$.
- If $|\overline{\mathcal{M}}^{(l)}| > \overline{N}^*$, set $\overline{N}^* = |\overline{\mathcal{M}}^{(l)}|$ and $\Delta^* = \Delta^{(l)}$.
- If $\underline{\mathcal{M}}^{(l)} = \varnothing$ or $\overline{\mathcal{M}}^{(l)} = \varnothing$, then go to step 4.
- If $\underline{\mathcal{M}}^{(l)} \neq \varnothing$ and $\overline{\mathcal{M}}^{(l)} \neq \varnothing$, then run the *"updating process"* as follows:

% Start of updating process %
a: For user $i \in \underline{\mathcal{M}}^{(l)}$, set $\alpha_i^{(l+1)} = \alpha_i^{(l)}$.
b: For user $i \in \overline{\mathcal{M}}^{(l)}$, set $\alpha_i^{(l+1)}$ so that
   i) $\alpha_i^{(l)} > \alpha_i^{(l+1)} \ge 0$ if $\alpha_i^{(l)} > 0$.
   ii) $\alpha_i^{(l+1)} = \alpha_i^{(l)}$ if $\alpha_i^{(l)} = 0$.
% End of updating process %

3: Increase l and go back to step 2 until there is no update request for $\Delta^{(l)}$.

4: Set $\Delta := \Delta^*$ and run the HPC algorithm until convergence.

We exploit this fact to develop the HPC adaptation algorithm, which is described in Algorithm 3. Let $\overline{\mathcal{M}}^{(l)}$ and $\underline{\mathcal{M}}^{(l)}$ be the sets of supported and nonsupported users in iteration l, respectively. We use $\overline{N}^*$ to keep the number of supported users during the course of the algorithm. In each iteration, each user i will run the proposed HPC algorithm and slowly update α_i based on the achieved equilibrium. All data users i initially set α_i to be a sufficiently large value so that we reach the first equilibrium that favors strong users. For each nonsupported user i, we maintain its parameter α_i to keep its power updating process stable. In addition, user i can save its parameter α_i at the instant when the number of nonsupported users decreases and reloads this value later as the global update process terminates (in step 4). This helps to prevent potential users from reducing their transmit powers unnecessarily. Discussion about the general *"updating process"* is given in Algorithm 3 whose specific design will be presented in Algorithm 4. In fact, detailed design of the *"updating process"* can be done to meet specific design objectives. However, if the *"updating process"* is designed in such a way that all parameters α_i of supported users tend to zero if all nonsupported users cannot be saved, then the proposed HPC adaptation algorithm can achieve the following desirable performance.

Theorem 3: Let $\overline{N}_{\mathsf{HPC}}$ and $\overline{N}_{\mathsf{TPC}}$ be the numbers of supported users due to the proposed HPC adaptation algorithm (i.e., Algorithm 3) and the TPC algorithm, respectively. Then, we have the following.

1) $\overline{N}_{\mathsf{HPC}} \geq \overline{N}_{\mathsf{TPC}}$.
2) If the network is feasible (i.e., all SINR requirements can be fulfilled by the TPC algorithm), then all users achieve their target SINRs by using the HPC adaptation algorithm, and there exist feasible users who achieve SINRs higher than the target values under the HPC adaptation algorithm.

Proof: The proof is given in Appendix E. ■

This theorem implies that the proposed HPC adaptation algorithm achieves better performance than the traditional TPC algorithm for both infeasible and feasible systems. Specifically, the HPC adaptation algorithm can support at least the same number of users and achieve higher SINRs and, therefore, higher total throughput compared with those due to the TPC algorithm.

B. Practical Algorithm Design

We propose a practical design of the *"updating process"* for HPC adaptation presented in Algorithm 3. In fact, this process attempts to convert nonsupported users into as many supported users as possible by decreasing effective interference levels of nonsupported users below threshold value R_i^{th}. Note that the interference received by a user is contributed by all other users in the network where intracell interference forms a significant portion of the total interference. Exploiting this fact, we propose to update α_i in Δ in a process consisting of both local and global updates, i.e., locally varying α_i within each cell and globally varying α_i when there are cells whose users are not supported after performing the local updating process.

Let $\mathcal{M}_k$ be the set of users associated with BS k at the equilibrium, i.e., $\mathcal{M}_k = \{j|b_j \equiv k\}$. In addition, let $\overline{\mathcal{M}}_k$ and $\underline{\mathcal{M}}_k$ be the sets of supported and nonsupported users in cell k, respectively. Therefore, we have $\overline{\mathcal{M}}_k \cup \underline{\mathcal{M}}_k = \mathcal{M}_k$. Let $\Delta_k = \{\alpha_i | i \in \mathcal{M}_k\}$. For given values of α_i in Δ_k, if the SINRs of users in $\mathcal{M}_k$ are all satisfied or unsatisfied, then we do not change $\alpha_j (j \in \mathcal{M}_k)$ further in the local updating process. If there are both supported and nonsupported users in a particular cell, we propose to reduce the power of the potential users to improve the SINRs of nonsupported users. In particular, we can calculate the reduction ratio of effective interference required for the weak user $i (i \in \underline{\mathcal{M}}_k)$ as follows:

$$\beta_i = \frac{R_i(p) - R_i^{\mathsf{th}}}{R_i(p)}, \qquad i \in \underline{\mathcal{M}}_k. \tag{15}$$

Consultation of $R_i(p)$ in (2) suggests that to assist user $i (i \in \underline{\mathcal{M}}_k)$ in reducing $R_i(p)$ by factor β_i, all potential users must reduce their transmit power by a factor of at least β_i. In addition, we need to assist the weakest user who has the highest effective interference in achieving its target SINR. The reduction ratio of effective interference for this weakest user can be expressed as

$$\beta_{\mathsf{max}}^k = \max_{i \in \underline{\mathcal{M}}_k} \beta_i. \tag{16}$$

On the other hand, each potential user j must maintain its SINR requirement, i.e., its new SINR must not be less than its target SINR. Hence, its new transmit power level must not be less than $R_j(p)\hat{\gamma}_j$. Therefore, the expected transmit power of potential user j in cell k can be written as

$$p_j^{\mathsf{exp}} = \max\left\{p_j\left(1 - \beta_{\mathsf{max}}^k\right), R_j(p)\hat{\gamma}_j\right\}, \quad j \in \overline{\mathcal{M}}_k. \tag{17}$$

Using this result, we can calculate parameter α_j for potential user j after using (13) and performing some manipulations as

$$\alpha_j = g\left(p_j^{\mathsf{exp}}\right) = \frac{p_j^{\mathsf{exp}} - R_j(p)\hat{\gamma}_j}{\xi_j R_j(p)^{\frac{x}{x-1}} - p_j^{\mathsf{exp}}}, \quad j \in \overline{\mathcal{M}}_k. \tag{18}$$

This local updating process for each cell k is run until all users meet their SINR requirements or there is no potential user in this cell. A cell is called unsatisfied if it still contains nonsupported users (and there is no potential user). If there exist unsatisfied cells, the BS of each unsatisfied cell will send a "warning message" through the backhaul to seek global assistance. Then, the global updating process will be performed as soon as a "warning message" is broadcast. In this global updating process, all potential users i will update their parameters α_i to assist the unsatisfied cells. The global updating process can be run in parallel with the local updating process, i.e., potential user i may reduce α_i twice in one updating step.

We propose a practical *"updating process,"* which is described in Algorithm 4. Let $\overline{\mathcal{M}}_k^{(l)}$ and $\underline{\mathcal{M}}_k^{(l)}$ be the sets of supported and nonsupported users in cell k and iteration l, respectively. In this proposed algorithm, each user attempts to assist other users in achieving their SINR targets while achieving a high SINR for itself. After obtaining the equilibrium in each iteration, each user i whose SINR is lower than the target $\hat{\gamma}_i$ at the equilibrium will maintain its parameter α_i. The supported or potential users who have their α_i greater than zero will update their α_i according to (18) or scale down their α_i further by a scaling factor of $\varsigma (\varsigma > 1)$ if they receive "warning messages." A "warning message" from a particular cell includes the number of nonsupported users of the cell. Hence, each user knows whether its reduction of α_i is helpful or not. Each user stores its parameter α_i at the instant when the number of nonsupported users decreases and reloads this value after the global update process terminates. The global process will terminate as all users attain their target SINRs or all the supported users reduce their parameters α_i to zero.

Algorithm 4 EXEMPLIFIED DESIGN OF UPDATING PROCESS

1: Run the following *"updating process"* at each cell $k (k = 1, 2, \dots, K)$ in iteration l (only for data users):

- If $\underline{\mathcal{M}}_k^{(l)} \neq \varnothing$, then consider the following cases:
  - a: For $i \in \underline{\mathcal{M}}_k^{(l)}$, set $\alpha_i^{(l+1)} = \alpha_i^{(l)}$.
  - b: For $i \in \overline{\mathcal{M}}_k^{(l)}$, perform the following:
    - b1: Set $\alpha_i^{(l+1)} = g(p_j^{\text{exp}})$ if BS k does not receive any "warning massage."
    - b2: Set $\alpha_i^{(l+1)} = g(p_j^{\text{exp}})/\varsigma$ if BS k receives a "warning massage."
  - c: If $\alpha_i^{(l)} = 0, \ \forall i \in \overline{\mathcal{M}}_k^{(l)}$, then BS k sends a "warning message" to other BSs in the network.
- If $\underline{\mathcal{M}}_k^{(l)} = \varnothing$, consider the following cases:
  - a: If BS k receives a "warning message," then set $\alpha_i^{(l+1)} = \alpha_i^{(l)}/\varsigma$ for $i \in \overline{\mathcal{M}}_k^{(l)}$.
  - b: If BS k does not receive any "warning message," then set $\alpha_i^{(l+1)} = \alpha_i^{(l)}$ for $i \in \overline{\mathcal{M}}_k^{(l)}$.

C. *Application to Two-Tier Macrocell–Femtocell Networks*

1) Two-Tier Macrocell–Femtocell Networks: We consider a two-tier network where M_f FUE devices served by K FBSs are underlaid with one macrocell serving M_m MUE devices. We denote the sets of MUE and FUE devices by $\mathcal{M}_m \triangleq \{1, \dots, M_m\}$ and $\mathcal{M}_f \triangleq \{M_m + 1, \dots, M_m + M_f\}$, respectively, and the set of all users is $\mathcal{M} = \mathcal{M}_m \cup \mathcal{M}_f$. We assume that FUE devices are only served by the BS of its cell. However, MUE devices are allowed to connect to any BS in the network. All users in this heterogeneous network are assumed to employ the proposed HPC scheme. In addition, by allowing users to adjust appropriate PC parameters, we can provide useful mechanisms for FBSs to control the spectrum access from associated MUE devices, which enables the hybrid access design for the underlying two-tier network.

2) Control MUE Devices' Access at Femtocells: Recall that MUE devices are allowed to connect to any FBS to enhance their performance and to reduce the interference to other users in the network. However, MUE devices' connections to a particular femtocell without appropriate control may significantly degrade the throughput of FUE devices in the underlying femtocell, which is not desirable from the FUE devices' viewpoint. To resolve this issue, we propose a hybrid access strategy to control the access of MUE devices. Toward this end, we assume that each FUE i has two target SINRs, e.g., $\hat{\gamma}_i^{[1]}$ and $\hat{\gamma}_i^{[2]}$, where $\hat{\gamma}_i^{[1]} < \hat{\gamma}_i^{[2]}$. Then, FUE devices can attempt to reach the higher target SINRs as long as the network condition allows. Otherwise, FUE devices seek to maintain at least the lower target SINRs when it is possible.

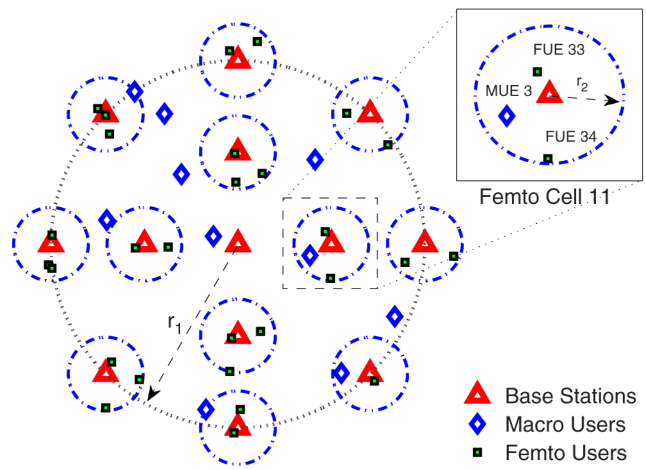

Fig. 2. Simulated two-tier macrocell–femtocell network.

Each FUE i can adaptively track the higher target SINR by taking the following procedure. It employs the proposed HPC adaptation algorithm to achieve the lower target SINR $\hat{\gamma}_i^{[1]}$ in the first phase. If this is successfully fulfilled, it attempts to achieve the higher target SINR $\hat{\gamma}_i^{[2]}$ in the second phase. Toward this end, FUE i simply sets parameters ξ_i and R_i^{th} by using the higher target SINR. Then, it runs the proposed HPC adaptation algorithm again. As FUE devices at a particular femtocell attempt to achieve a higher target SINR, MUE devices associated with the underlying femtocell have to decrease their transmit powers and, therefore, achieve lower SINRs. Hence, the proposed HPC adaptation algorithm enables us to attain flexible spectrum sharing between FUE and MUE devices at each femtocell.

V. NUMERICAL RESULTS

We present illustrative numerical results to demonstrate the performance of the proposed algorithms. The network setting and user placement for our simulations are shown in Fig. 2, where MUE and FUE devices are randomly located inside circles of radii of $r_1 = 1000$ m and $r_2 = 50$ m, respectively. Assume that there are heavy walls at the boundaries of the femtocells. We fix $M_m = 10$ except for the results given in Fig. 4 and randomly choose the number of FUE devices in each femtocell from 1 to 3. Then, eight users of either tier are set as voice users randomly. Channel power gain h_{ij} is chosen according to path loss $L_{ij} = A_i \log_{10}(d_{ij}) + B_i + C \log_{10}(f_c/5) + W_l n_{ij}$, where d_{ij} is the distance from user j to BS i; (A_i, B_i) values are set as (36, 40) and (35, 35) for MBSs and FBSs, respectively; $C = 20$, $f_c = 2.5$ GHz; W_l is the wall-loss value; and n_{ij} is the number of walls between BS i and user j. Other parameters are set as follows (unless stated otherwise): $x = 0.5$; processing gain $G = 128$; maximum power $\bar{p}_j = 0.01$ W, $\forall j \in \mathcal{M}$; noise power $\eta_i = 10^{-13}$ W, $\forall i \in \mathcal{K}$. We set n_{ij} equal to the number of cell boundaries that the corresponding signal traverses and the wall-loss value to $W_l = 12$ dB (except in Fig. 12). The SINR presented in each figure is either in linear or decibel scale, which is stated below each relevant figure in this section.

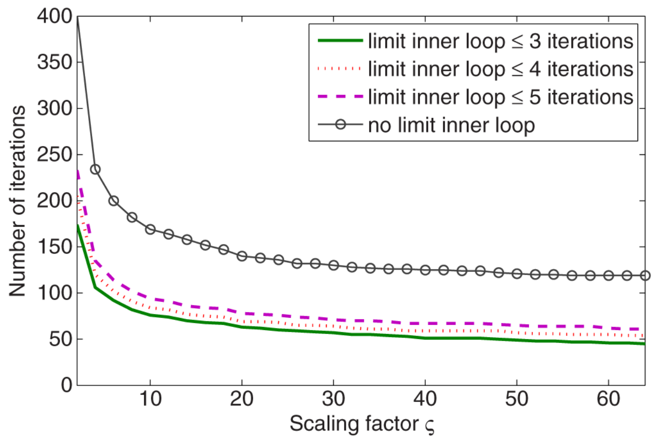

Fig. 3. Number of iterations versus scaling factor ς.

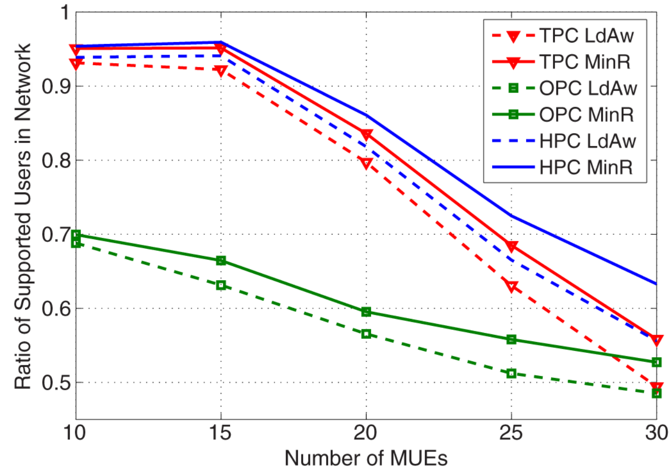

Fig. 4. Ratio between the number of supported users and the number of all users under different BSA schemes for target SINRs of 8 (in linear scale).

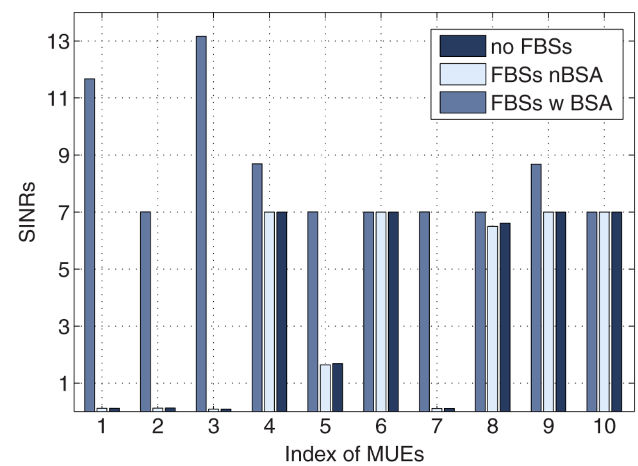

Fig. 5. SINRs of MUE devices with and without FBS association (closed versus hybrid access) for target SINRs of 7 (in linear scale). The "index of MUE devices" in the horizontal axis indicates individual MUE devices in the simulation.

In Algorithm 3, we have to run the HPC power updates given in (14) in the inner loop before updating parameters $\alpha_i^{(l)}$ in each iteration of the outer loop if there exists any user who cannot achieve their target SINRs (i.e., the set $\underline{\mathcal{M}}^{(l)} \neq \emptyset$). We can limit the number of HPC power updates (iterations) to smaller than a predetermined limit instead of waiting for HPC power updates to fully converge to improve the convergence time of Algorithm 3. Moreover, we can also improve the convergence time of Algorithm 3 by appropriately setting scaling factor ς in Algorithm 4. In fact, using a larger scaling factor ς enables us to relieve network congestion more quickly but may degrade the throughput performance. To study the impacts of these possible parameter settings on the convergence time of the proposed algorithm, we show the number of required iterations versus different values of ς in Fig. 3, where the number of HPC power updates (iterations) is limited to three, four, five, and "no limit," which are indicated as "limit inner loop" in this figure. The average total throughput remains almost unchanged for the considered values of ς and number of HPC power updates. We do not present the throughput variation here due to the space constraint. Fig. 3 confirms that we can significantly improve the convergence time of Algorithm 3 by choosing sufficiently large ς (e.g., $\varsigma > 10$) and a small number of HPC power updates (e.g., three or five HPC power updates are sufficient). Learning from these studies, we set $\varsigma = 16$ and the number of HPC power updates equal to five to obtain all other results presented in the following.

In Fig. 4, we show ratios between the number of supported users and the total number of users for the following schemes: dynamic BSA and fixed load-aware BSA strategies (i.e., BSA in Algorithm 1 and Algorithm 2, respectively) with different PC strategies, namely, TPC, OPC, and HPC adaptation algorithms (i.e., Algorithm 3 with the updating process given in Algorithm 4). In this figure, we increase the number of MUE devices from 10 to 30. For the fixed BSA (i.e., in the decomposed BSA and PC design), we use the average channel power gains based only on path loss. For the dynamic BSA (i.e., in the joint BSA and PC design), we determine BSA results based on instantaneous channel power gains comprising both path loss and Rayleigh fading, which is represented by an exponentially distributed random variable with the mean value of 1. Results corresponding to fixed and dynamic BSAs are indicated by "LdAw" and "MinR," together with the corresponding PC schemes (i.e., TPC, OPC, and HPC) in this figure, respectively. As shown in this figure, the proposed HPC scheme outperforms existing TPC and OPC schemes under both joint and decomposed BSA and PC designs. Moreover, the dynamic BSA proposed in Algorithm 1 achieves better performance than the fixed BSA for all three PC strategies. In addition, the performance gap between joint and decomposed BSA and PC designs becomes quite significant when the number of MUE devices is sufficiently large. Therefore, the joint BSA and PC proposed in Algorithm 1 will be employed to obtain other results in this section.

In Fig. 5, we present the SINRs of MUE devices for three cases: 1) There is no femtocell; 2) there are femtocells, but MUE devices are not allowed to connect with FBSs (i.e., closed access); and 3) there are femtocells, and MUE devices are allowed to connect with FBSs (i.e., hybrid access). Results for these three cases are indicated as "no FBSs," "FBSs nBSA," and "FBSs wBSA" in this figure, respectively. The target SINRs of all users are set to 7. It can be observed that the SINRs of MUE devices do not decrease much with the deployment

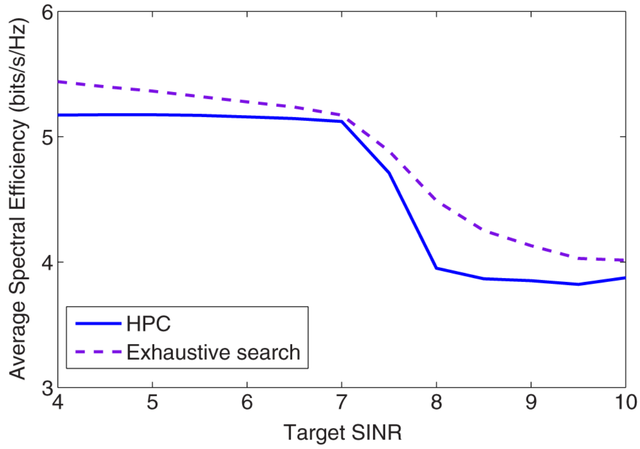

Fig. 6. Comparison of the throughput achieved by Algorithm 3 and the exhaustive search algorithm. (The target SINR is in linear scale.)

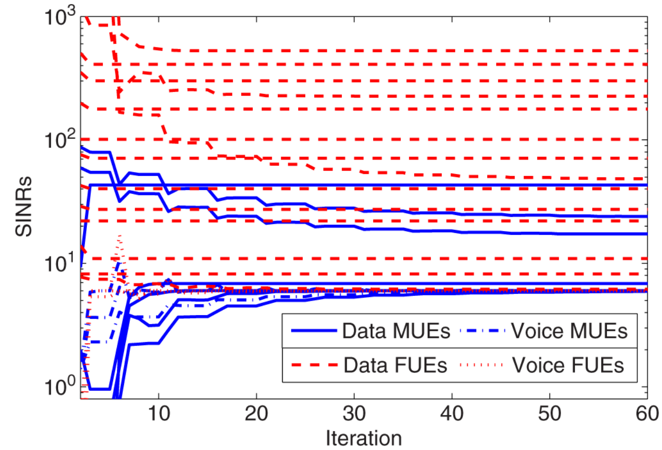

Fig. 7. SINRs of MUE and FUE devices for target SINRs of 6 (in linear scale).

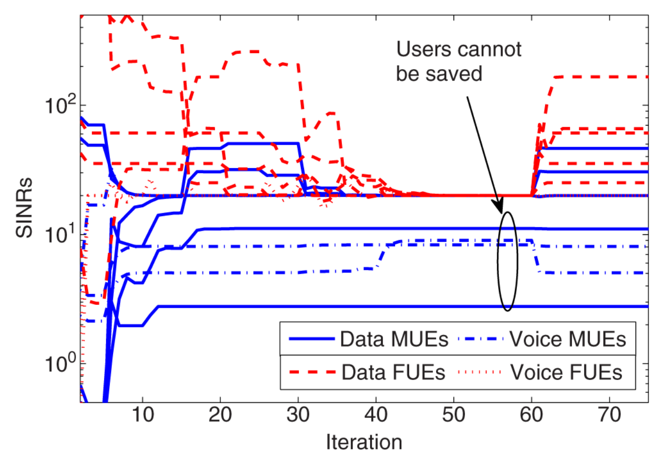

Fig. 8. SINRs of all users for target SINRs of 20 (in linear scale).

of femtocells. This is due to the fact that the large wall loss prevents femtocells from creating large cross-tier interference to MUE devices. Moreover, by allowing MUE devices to connect with nearby FBSs, we can significantly increase their SINRs. Moreover, as shown in Fig. 5, the average SINRs per one FUE for the cases with and without FBS association are approximately 1267 and 39, respectively. These results confirm the great benefits of the hybrid access over the closed access in the two-tier macrocell–femtocell network in improving the SINRs of both MUE and FUE devices.

The operating point achieved by Algorithm 3 may not lie on the Pareto-optimal boundary of the capacity region. Therefore, there can exist a different point that strictly dominates this operating point. To understand the throughput gap between our solution and a "throughput optimal" solution, we have developed an exhaustive search algorithm that aims to reach a more throughput-efficient point on the Pareto-optimal boundary of the capacity region. This exhaustive search algorithm performs two key steps in each iteration, which are described in the following. *In the first step*, we consider all possible pairs of users (i, j) and scale their SINRs as $\Gamma_i^{(k)} = \Gamma_i^{(k)}/\theta$ and $\Gamma_j^{(k)} = \theta\Gamma_j^{(k)}$, where scaling factor $\theta > 0$ is searched so that the maximum throughput (i.e., spectral efficiency) can be achieved and all supported users are still supported after the SINR updates for the users (i, j). The operation in this step aims to increase the total throughput while maintaining the set of supported users. *In the second step*, we find exactly one user where the total throughput can be increased the most as we increase the SINR of this user while keeping infeasibility of the resulting set of SINRs. Then, the SINR of the chosen user is set to the maximum value. These two steps are performed in each iteration of the search algorithm until convergence.

In Fig. 6, we compare the throughput achieved by Algorithm 3 and the exhaustive search algorithm. As shown in this figure, the exhaustive search algorithm achieves higher throughput than that achieved by Algorithm 3. Note, however, that the exhaustive search algorithm relies on the operating point achieved by Algorithm 3 for initialization. In addition, it is the centralized algorithm and is based on exhaustive search. Fig. 6 also reveals that the throughput gap between the two algorithms is pretty small for low target SINR values where the QoS constraints of all users can be supported. This confirms the efficacy of our proposed low-complexity algorithm.

Figs. 7 and 8 show the convergence of the HPC adaptation algorithm when we set $\hat{\gamma}_i = 6$ and 20 for all users, respectively. Fig. 7 confirms that the HPC adaptation algorithm converges to an equilibrium for which some users exactly achieve their target SINRs while others attain SINRs higher than their target values. This set of results corresponds to the feasible system where all SINR requirements can be supported. This figure shows that the proposed HPC adaptation algorithm not only attains all SINR requirements but enables strong users to achieve high throughput performance as well. The results in Fig. 8 correspond to an infeasible system where we cannot attain all SINR requirements. The dynamics of the HPC adaptation algorithm demonstrated in this figure can be interpreted as follows. Some potential users i attempt to reduce their parameters α_i to zero to assist nonsupported users to achieve their target SINRs initially. As soon as potential users recognize that there are still nonsupported users, they reload the last α^* values (i.e., in step 4 of Algorithm 3), which result in improvements for their SINRs.

In Fig. 9, we show the fairness achieved by the OPC, TPC, and HPC adaptation algorithms by showing the highest and

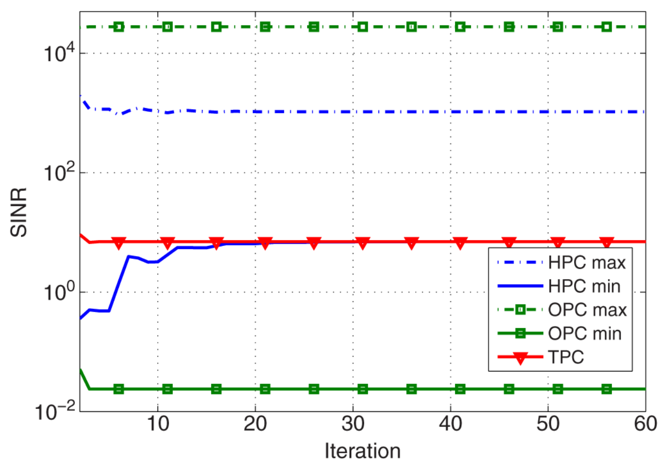

Fig. 9. Maximum and minimum SINRs achieved by OPC, TPC, and HPC adaptation schemes for target SINRs of 7 (in linear scale).

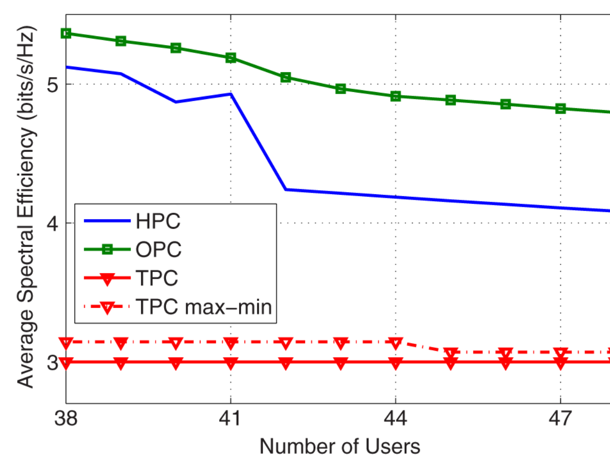

Fig. 11. Average spectral efficiency versus number of all users for target SINRs of 7 (in linear scale).

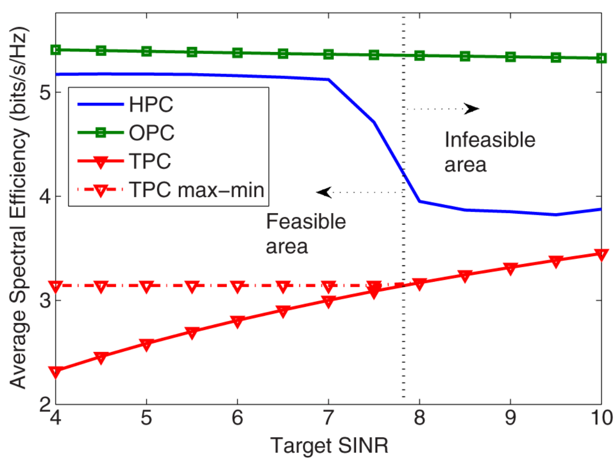

Fig. 10. Average spectral efficiency versus target SINR (in linear scale).

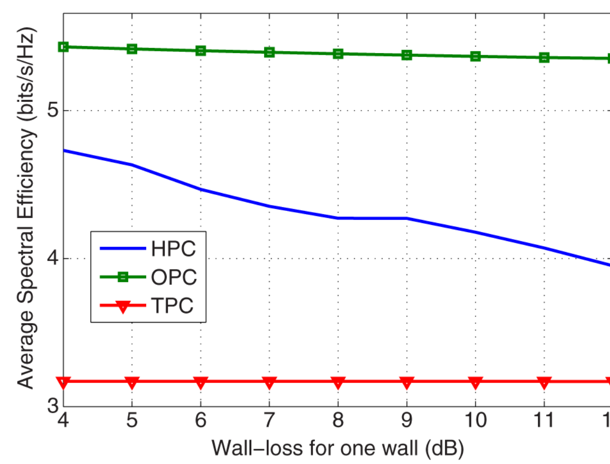

Fig. 12. Average spectral efficiency versus the wall-loss value for target SINRs of 8 (in linear scale).

lowest SINRs of all users for different strategies (indicated as X min and X max in this figure, where X represents the corresponding PC strategy). In the TPC scheme, all users reach the same target SINR. In contrast, the OPC scheme results in very different SINRs at the equilibrium where the strongest user attains a very high SINR, whereas the weak users achieve SINRs that are very close to zero. Therefore, the OPC scheme tends to allocate more resources to strong users to achieve high overall throughput at the cost of significantly penalizing weak users. This figure shows that our proposed HPC adaptation algorithm allows the weak users to achieve their target SINRs while enabling strong users to settle at higher SINRs. Therefore, our proposed algorithm strikes to balance between achieving required QoS while exploiting multiuser diversity to enhance the system throughput, which would be very desirable for data-driven wireless systems.

Figs. 10 and 11 show the average spectral efficiency (i.e., throughput) achieved by different schemes versus the target SINR and the number of all users, respectively. Here, the spectral efficiency of user i is calculated as $\log_2(1+\Gamma_i)$ (in bits per second per hertz). To obtain the results in these two figures, we randomly generate user locations and obtain results for this fixed topology. For the results in Fig. 11, we sequentially add more FUE devices to obtain different points on each curve. For reference, we also present the average spectral efficiency of a TPC max–min scheme in which we slowly increase the target SINR for all users as long as the system is still feasible under the TPC scheme. (This scheme is denoted as TPC max–min in these two figures.) As shown in these figures, our HPC adaptation algorithm attains higher average spectral efficiency than both traditional TPC and TPC max–min schemes but lower average spectral efficiency than the OPC algorithm. In particular, when the network is lowly loaded ($\hat{\gamma}_i < 8$ in Fig. 10), our proposed algorithm attains much higher spectral efficiency than the traditional TPC and TPC max–min schemes. Moreover, when the network load is higher ($\hat{\gamma}_i \geq 8$), the gaps between our proposed algorithm and the two TPC schemes become smaller. This is because Algorithm 3 attempts to maintain the target SINRs for all users as the TPC schemes.

In Fig. 12, we study the impact of wall-loss value W_l (therefore, the path loss L_{ij}) on the average spectral efficiency of different schemes. This figure shows that the average spectral efficiency of the TPC scheme remains unchanged, whereas that of the OPC scheme slightly decreases with the increasing wall-loss value W_l. This is because the system load under consideration is light; hence, the number of users that can be

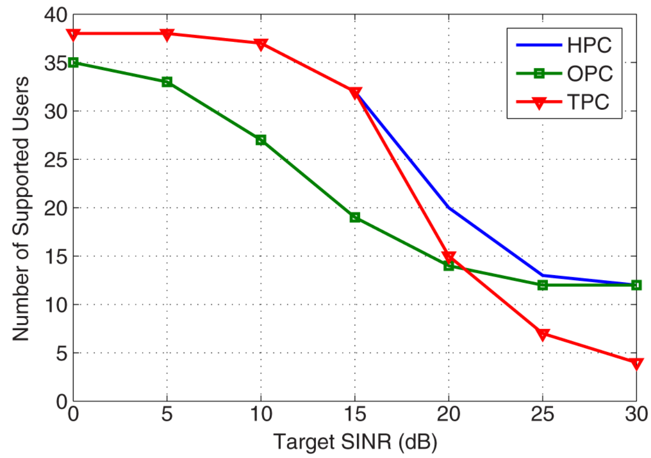

Fig. 13. Number of supported users in HPC, TPC, and OPC schemes versus target SINR (in decibel scale).

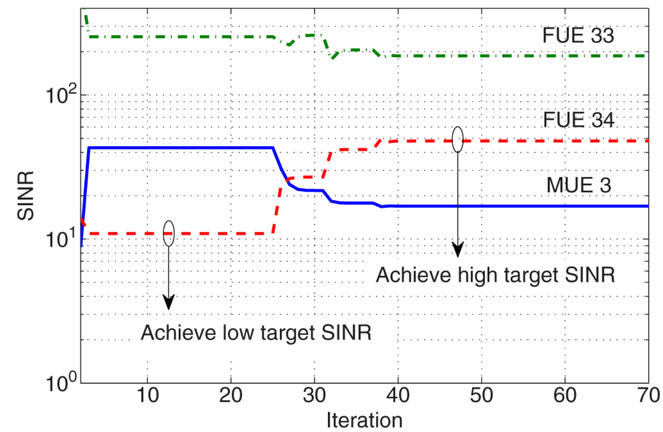

Fig. 14. SINR of users in femtocell 11 as FUE 34 attempts to achieve the higher target SINR (in linear scale).

supported by the TPC scheme remains the same for different values of W_l. Moreover, the OPC scheme allocates most capacity to a few strong users who achieve very high SINRs and contribute a significant fraction of the total throughput. As a result, the throughput contribution from these strong users only slightly decreases with increasing wall-loss value W_l due to the concavity characteristic of the throughput formula. Finally, the spectral efficiency of the proposed HPC scheme decreases with W_l; however, it consistently achieves better performance than the TPC scheme.

Fig. 13 shows the number of supported users for different schemes. It is shown that our proposed HPC adaptation algorithm can maintain the SINR requirements for the larger number of users compared with the TPC and OPC algorithms. In particular, our proposed algorithm can support roughly the same number of users as the TPC algorithm for low values of target SINRs. However, our proposed algorithm performs better than the TPC scheme for higher target SINRs. In fact, almost all users utilize the maximum power under the TPC scheme at high target SINRs, which prevent all of them from achieving their target SINRs. On the other hand, in our proposed scheme, nonsupported users tend to use as small transmit power as possible (i.e., high values of α_i). This enables more users to attain their target SINRs. The figure also shows that the proposed HPC adaptation algorithm performs better than the OPC scheme for all values of the target SINR.

In Fig. 14, we show the SINR dynamics in femtocell 11 under the association of one MUE (MUE 3). This femtocell and the corresponding users are indicated in Fig. 2. Specifically, FUE devices attempt to achieve the low target SINR initially, which is equal to 5. As this SINR requirement is fulfilled, FUE 34 is not "happy" and attempts to achieve a higher target SINR of 50. This figure shows that the proposed HPC adaptation algorithm allows FUE 34 to achieve this QoS goal where it gradually increases its power to reach the new target SINR. This, in turn, results in the decrease in SINR of MUE 3 and the other FUE (FUE 33). This figure, therefore, confirms that our proposed algorithm allows FUE devices to achieve flexible spectrum sharing with associated MUE devices, which is a desirable feature to realize the hybrid access mode at femtocells. FUE devices 3, 33, and 34 are indicated in Fig. 2.

VI. Conclusion

We have developed a general distributed BSA and PC algorithm and demonstrated how it can be utilized to design an efficient hybrid spectrum access strategy for two-tier macrocell–femtocell networks. The BSA mechanism integrated in this algorithm aims to minimize user effective interference. In addition, a novel HPC scheme has been proposed, which can be used in this algorithm. We have proved the convergence of the proposed algorithm by using the 2.s.s. function approach. We have also developed an efficient adaptation mechanism for the proposed algorithm so that users can achieve their SINR requirements whenever possible while we can exploit multiuser diversity to increase the system throughput. Numerical results were presented to demonstrate the desirable performance of the proposed algorithms.

Appendix A
Proof of Theorem 1

According to Lemma 1, a PC algorithm converges if the corresponding puf is 2.s.s. Hence, we can prove Theorem 1 by showing that puf $J^\circ(p) = [J_1^\circ(p), \ldots, J_M^\circ(p)]$ is a 2.s.s. function with respect to p, where $J_i^\circ(p) = J_i'(R_i^\circ(p))$, and $R_i^\circ(p) = \min_{k \in D_i} R_i(p,k), \forall\, i \in \mathcal{M}$. We will prove $J_i'(R_i^\circ(p))$ in the following.

Recall that we have assumed $J'(R(p))$ to be 2.s.s. with respect to $R(p)$, where $J(p) = J'(R(p))$. Therefore, $J'(R^\circ(p))$ is a 2.s.s. function with respect to $R^\circ(p)$. Consequently, the proof is equivalent to showing the following statement: If $(1/a)p \le p' \le ap$, then we have

$$\frac{1}{a} R_i^\circ(p) \le R_i^\circ(p') \le a R_i^\circ(p) \qquad \forall\, i \in \mathcal{M}. \tag{19}$$

We will prove (19) by using contradiction. Let b_i and b_i' be the BSs chosen by user i corresponding to $R_i^\circ(p)$ and $R_i^\circ(p')$, respectively. From our assumption $(1/a)p \le p' \le ap$ for $a > 1$, we have $(1/a)p_j \leq p'_j \leq ap_j, \forall j \in \mathcal{M}$. Performing some simple manipulations, we can obtain

$$(1/a)R_i(p, b'_i) \leq R_i^\circ(p') \leq aR_i(p, b'_i)$$
$$(1/a)R_i^\circ(p) \leq R_i(p', b_i) \leq aR_i^\circ(p). \tag{20}$$

Now suppose that (19) is not satisfied if $(1/a)p \leq p' \leq ap$. Then, we must have $(1/a)R_i^\circ(p) > R_i^\circ(p')$ or $R_i^\circ(p') > aR_i^\circ(p)$. Let us consider these possible cases in the following.

- If $(1/a)R_i^\circ(p) > R_i^\circ(p')$, then using the results in (20) yields
$$R_i^\circ(p) > R_i(p, b'_i). \tag{21}$$
- Similarly, if $R_i^\circ(p') > aR_i^\circ(p)$, then using the results in (20) yields
$$R_i^\circ(p') > R_i(p', b_i). \tag{22}$$

Both (21) and (22) indeed result in contradiction to the definition of $R_i^\circ(p)$ and $R_i^\circ(p')$, which is given in (7). Hence, we have $(1/a)R_i^\circ(p) \leq R_i^\circ(p') \leq aR_i^\circ(p)$ if $(1/a)p \leq p' \leq ap$, which completes the proof of the theorem.

Appendix B
Proof of Lemma 2

We can rewrite the payoff function in (10) as

$$U_i^{(1)}(p) = \Gamma_i^x - \lambda_i p_i = \left(\frac{p_i}{R_i(p)}\right)^x - \lambda_i p_i. \tag{23}$$

Taking the first and second derivatives of this payoff function with respect to p_i yields

$$\frac{\partial U_i^{(1)}}{\partial p_i} = \frac{x}{R_i(p)}\left(\frac{p_i}{R_i(p)}\right)^{x-1} - \lambda_i \tag{24}$$
$$\frac{\partial^2 U_i^{(1)}}{\partial p_i^2} = \frac{x(x-1)}{R_i(p)^2}\left(\frac{p_i}{R_i(p)}\right)^{x-2}. \tag{25}$$

From (25), it can be easily verified that $\partial^2 U_i^{(1)}/\partial p_i^2 < 0$ for $0 < x < 1$, which implies that $U_i^{(1)}(p)$ is a concave function. Therefore, the best response can be obtained by setting the first derivative in (24) to zero. After some manipulations, we can obtain the best response corresponding to payoff function $U_i^{(1)}(p)$ as follows:

$$p_i = R_i(p)^{\frac{x}{x-1}}(\lambda_i/x)^{\frac{1}{x-1}} = \xi_i R_i(p)^{\frac{x}{x-1}} \tag{26}$$

where the second relationship in (26) holds for $\xi_i = (\lambda_i/x)^{1/x-1}$. Moreover, it can be also verified that the best response obtained in (26) is exactly that corresponding to payoff function $U_i^{(2)}(p)$. Therefore, we have proved the lemma.

Appendix C
Proof of Theorem 2

Here, we will show that puf $I^H(p) = [I_1^H(p), \ldots, I_M^H(p)]$ in (13) is 2.s.s. with respect to p. Note that if $(1/a)p \leq p' \leq ap$ for a given $a > 1$, then from the definition of $R_i(p)$ in (2), we have

$$(1/a)R_i(p) \leq R_i(p') \leq aR_i(p) \qquad \forall i \in \mathcal{M}. \tag{27}$$

Let $I_i'^H(R_i(p)) = \min\{\bar{p}_i, I'_i(R_i(p))\}$, which is a function of $R_i(p)$. Note that we have $I_i^H(p) = I_i'^H(R_i(p))$. To prove that $I^H(p)$ is 2.s.s. with respect to p, we can equivalently show that $I_i'^H(R_i(p))$ is 2.s.s. with respect to $R_i(p)$.

Since $x/(x-1) < 0$ for $0 < x \leq 1/2$, it can be verified that $I'_i(R_i(p))$ is convex with respect to $R_i(p) > 0$ and $\lim_{R_i(p)\to\{0,\infty\}} I'_i(R_i(p)) = \infty$. Hence, there are, at most, two values of $R_i(p)$ that satisfy $I'_i(R_i(p)) = \bar{p}_i$.

If there is no or only one such intersection point, we have $I'_i(R_i(p)) \geq \bar{p}_i$ or $I_i'^H(R_i(p)) = \bar{p}_i$. For both cases, we have $(1/a)I_i'^H(R_i(p)) < I_i'^H(R_i(p')) < aI_i'^H(R_i(p))$ since $a > 1$.

If there are two intersection points, let R_i^l and R_i^u be the values of $R_i(p)$ at these two points, where $R_i^l < R_i^u$. Moreover, the puf of the HPC algorithm in these cases must satisfy

$$I_i'^H(R_i(p)) = \begin{cases} I'_i(R_i(p)), & \text{if } R_i^l \leq R_i(p) \leq R_i^u \\ \bar{p}_i, & \text{otherwise.} \end{cases} \tag{28}$$

Let us consider all possible cases in the following.

- If $\{R_i(p), R_i(p')\} \not\subset [R_i^l, R_i^u]$, then we have $I_i'^H(R_i(p)) = I_i'^H(R_i(p')) = \bar{p}_i$. Therefore, it is easy to obtain that $(1/a)I_i'^H(R_i(p)) < I_i'^H(R_i(p')) < aI_i'^H(R_i(p))$.
- If $\{R_i(p), R_i(p')\} \subset [R_i^l, R_i^u]$, then we have $I_i'^H(R_i(p)) = I'_i(R_i(p))$ and $I_i'^H(R_i(p')) = I'_i(R_i(p'))$. Let $r = R_i(p)/R_i(p')$, we have
$$I'_i(R_i(p)) = \frac{(1/r)^{\frac{x}{1-x}}\alpha_i\xi_i R_i(p')^{\frac{x}{x-1}} + r\hat{\gamma}_i R_i(p')}{\alpha_i + 1}. \tag{29}$$
In addition, using the results in (27) yields $\{1/r, r, (1/r)^{x/1-x}\} \subset [1/a, a]$ since $0 < x \leq 1/2$. Therefore, we have $(1/a)I'_i(R_i(p')) \leq I'_i(R_i(p)) \leq aI'_i(R_i(p'))$. Moreover, these inequalities hold if $x = 1/2$ and both $1/r$ and r are equal to a or $1/a$. Evidently, this cannot be satisfied since $a > 1$. Thus, if $0 < x \leq 1/2$, we must have
$$(1/a)I_i'^H(R_i(p)) < I_i'^H(R_i(p')) < aI_i'^H(R_i(p)). \tag{30}$$
- Finally, we have the remaining cases where $R_i(p)$ or $R_i(p') \in [R_i^l, R_i^u]$. We will only consider the case $R_i(p) \leq R_i^l \leq R_i(p') \leq R_i^u(*)$ since the proofs for other cases can be similarly done. Using the results in $(*)$ yields $I_i'^H(R_i(p)) = \bar{p}_i$, $I_i'^H(R_i^l) = \bar{p}_i = I'_i(R_i^l)$, and $I_i'^H(R_i(p')) = I'_i(R_i(p'))$. After performing some simple manipulations, we obtain
$$(1/a)R_i^l \leq R_i(p') \leq aR_i^l. \tag{31}$$
By applying the results in the previous case, we have $(1/a)I_i'^H(R_i^l) < I_i'^H(R_i(p')) < aI_i'^H(R_i^l)$. Since $I_i'^H(R_i^l) = I_i'^H(R_i(p)) = \bar{p}_i$, relationship (30) holds, which completes the proof for this case.

In summary, we have proved that our proposed puf is 2.s.s. for $0 < x \leq 1/2$. In fact, when $x > 1/2$, $(1/r)^{x/1-x}$ may not be in $[1/a, a]$. Hence, the puf might not be 2.s.s.

APPENDIX D
PROOF OF LEMMA 3

According to the power update rule of the HPC algorithm given in (13), the SINR of user i at convergence can be expressed as

$$\Gamma_i(p^*) = \frac{p_i^*}{R_i(p^*)} = \frac{\min\{\bar{p}_i, I_i(p^*)\}}{R_i(p^*)}. \quad (32)$$

Recall that $R_i^{\mathrm{th}} = \bar{p}_i/\hat{\gamma}_i$. Using the relationship in (14) yields

$$\Gamma_i(p^*) = \begin{cases} \hat{\gamma}_i \frac{R_i^{\mathrm{th}}}{R_i(p^*)}, & p_i^* = \bar{p}_i \\ \hat{\gamma}_i \frac{\alpha_i \left[R_i^{\mathrm{th}}/R_i(p^*)\right]^{\frac{1}{1-x}}+1}{\alpha_i+1}, & p_i^* < \bar{p}_i. \end{cases} \quad (33)$$

We can now utilize this result to prove the lemma. Specifically, let us consider the following cases.

1) If $R_i(p^*) > R_i^{\mathrm{th}}$, then we have $R_i^{\mathrm{th}}/R_i(p^*)$ and $[R_i^{\mathrm{th}}/R_i(p^*)]^{x/(1-x)}$ strictly less than 1 since $0 < x \le 1/2$. Hence, we have $\Gamma_i(p^*) < \hat{\gamma}_i$ if $p_i^* = \bar{p}_i$ in (33). We now prove statement 1 of the lemma for the remaining case. For $\alpha_i > 0$, we have $\Gamma_i(p^*) < \hat{\gamma}_i$ according to (33). In addition, for $\alpha_i = 0$, we have $I_i(p^*) = \hat{\gamma}_i R_i(p^*) > \hat{\gamma}_i R_i^{\mathrm{th}} = \bar{p}_i$. Hence, the HPC power update in (13) results in $p_i^* = \bar{p}_i$, which leads to $\Gamma_i(p^*) = \hat{\gamma}_i(R_i^{\mathrm{th}}/R_i(p^*)) < \hat{\gamma}_i$. Therefore, we have completed the proof for the first statement of the lemma.
2) If $R_i(p^*) \le R_i^{\mathrm{th}}$, then we have $R_i^{\mathrm{th}}/R_i(p^*) \ge 1$. Hence, using the results in (33) yields $\Gamma_i(p^*) \ge \hat{\gamma}_i$, $\forall\, \alpha_i \ge 0$, which completes the proof for the second statement of the lemma.
3) If $R_i(p^*) < R_i^{\mathrm{th}}$ and $\alpha_i = 0$, then we have $I_i(p^*) = \hat{\gamma}_i R_i(p^*) < \hat{\gamma}_i R_i^{\mathrm{th}} = \bar{p}_i$. As a result, we have $p_i^* = I_i(p^*)$ due to the HPC power update in (13), which results in $\Gamma_i(p^*) = \hat{\gamma}_i$. Inversely, if $\Gamma_i(p^*) = \hat{\gamma}_i$ and $R_i(p^*) < R_i^{\mathrm{th}}$, then we must have $p_i^* < \bar{p}_i$ [i.e., the second case in (33)]. Therefore, we have $p_i^* = I_i(p^*)$ and $\alpha_i = 0$, which completes the proof for the following statement of the lemma: If $R_i(p^*) < R_i^{\mathrm{th}}$, then $\Gamma_i(p^*) = \hat{\gamma}_i$ iff $\alpha_i = 0$. Note that we have $\Gamma_i(p^*) > \hat{\gamma}_i$ iff $\alpha_i > 0$ and $R_i(p^*) < R_i^{\mathrm{th}}$. We now prove the last statement of the lemma. Let us take the derivative of $I_i(p^*)$ with respect to α_i, then we have

$$\frac{\partial I_i(p^*)}{\partial \alpha_i} = R_i(p^*)\hat{\gamma}_i \frac{\left[R_i^{\mathrm{th}}/R_i(p^*)\right]^{\frac{1}{1-x}} - 1}{(1+\alpha_i)^2}. \quad (34)$$

This implies that if we have $R_i(p^*) < R_i^{\mathrm{th}}(p)$, then $I_i(p^*)$ decreases as α_i decreases.

APPENDIX E
PROOF OF THEOREM 3

Proof of Statement 1 in Theorem 3: If $\overline{N}_{\mathsf{HPC}} = |\mathcal{M}|$, we obviously have $\overline{N}_{\mathsf{HPC}} \ge \overline{N}_{\mathsf{TPC}}$. Therefore, we only need to consider the scenarios where there are one or several nonsupported users after running Algorithm 3. In these cases, as Algorithm 3 terminates (i.e., the end of step 3 in Algorithm 3), we have $\alpha_i^{(s)} = 0$, $\forall i \in \overline{\mathcal{M}}^{(s)}$, where s denotes the index of the last iteration. Let Δ^s be the set storing the values of $\alpha_i^{(s)}$, $i \in \mathcal{M}$, i.e., $\alpha_i^{(s)}$ is the value of α_i for user i in the last iteration (before we set them equal to $\alpha_i^* \in \Delta^*$ in step 4 of Algorithm 3). Since each potential user i reduces α_i until it reaches zero in Algorithm 3, we have $\alpha_i^{(s)} = 0, \forall\, i \in \overline{\mathcal{M}}^{(s)}$ and $\overline{N}_{\mathsf{HPC}} \ge \overline{N}^s$, where $\overline{N}^s$ is the number of supported users after running the HPC puf using Δ^s. Hence, $\overline{N}_{\mathsf{HPC}} \ge \overline{N}_{\mathsf{TPC}}$ holds if we can prove that

$$\overline{N}^s \ge \overline{N}_{\mathsf{TPC}}. \quad (35)$$

We will prove this in the following. Now, let p_s^* and p_{TPC}^* denote the power vectors at convergence, which are obtained by running the HPC algorithm with Δ^s and the TPC algorithm, respectively. Note that p_{TPC}^* is also the power vector at convergence if we run the TPC algorithm with initial powers set as $p_{\mathsf{TPC}}^{(0)} = p_s^*$. Let us consider this initial setting for the TPC scheme and investigate the following possible scenarios for a particular user i.

- If user i is supported, then we have $R_i(p_{\mathsf{TPC}}^{(0)}) = R_i(p_s^*) \le R_i^{\mathrm{th}}$, and $p_{i,\mathsf{TPC}}^{(0)} = p_{i,s}^* = \hat{\gamma}_i R_i(p_{\mathsf{TPC}}^{(0)})$ since $\alpha_i^{(s)} = 0$.
- If user i is nonsupported, then we have $R_i(p_{\mathsf{TPC}}^{(0)}) = R_i(p_s^*) > R_i^{\mathrm{th}}$ and $p_{i,\mathsf{TPC}}^{(0)} \le \bar{p}_i$.

Let us analyze the dynamics of the power updates due to the TPC algorithm. In particular, the power of each user is updated in the next iteration as follows.

- $p_{i,\mathsf{TPC}}^{(1)} = \hat{\gamma}_i R_i(p_{\mathsf{TPC}}^{(0)}) = p_{i,\mathsf{TPC}}^{(0)}$ if $R_i(p_{\mathsf{TPC}}^{(0)}) \le R_i^{\mathrm{th}}$.
- $p_{i,\mathsf{TPC}}^{(1)} = \bar{p}_i \ge p_{i,\mathsf{TPC}}^{(0)}$ if $R_i(p_{\mathsf{TPC}}^{(0)}) > R_i^{\mathrm{th}}$.

Therefore, we always have $p_{\mathsf{TPC}}^{(1)} \ge p_{\mathsf{TPC}}^{(0)}$. Moreover, it can be verified that $R_i(p_{\mathsf{TPC}}^{(n)}) \ge R_i(p_{\mathsf{TPC}}^{(n-1)})$, $\forall\, i \in \mathcal{M}$ if $p_{\mathsf{TPC}}^{(n)} \ge p_{\mathsf{TPC}}^{(n-1)}$. Using this result and consulting the TPC puf, we can show that $p_{\mathsf{TPC}}^{(n+1)} \ge p_{\mathsf{TPC}}^{(n)}$ if $p_{\mathsf{TPC}}^{(n)} \ge p_{\mathsf{TPC}}^{(n-1)}$. Hence, under the TPC algorithm, $p_{\mathsf{TPC}}^{(n)}$ will increase and converge to p_{TPC}^* if $p_{\mathsf{TPC}}^{(1)} \ge p_{\mathsf{TPC}}^{(0)}$. Therefore, we have

$$p_{\mathsf{TPC}}^* \ge p_{\mathsf{TPC}}^{(0)} = p_s^*. \quad (36)$$

From this, we can achieve $R_i(p_{\mathsf{TPC}}^*) \ge R_i(p_s^*)$, $\forall\, i \in \mathcal{M}$. Hence, if user i is a supported user under the TPC algorithm, $R_i(p_{\mathsf{TPC}}^*)$ must not be greater than R_i^{th}. As a result, $R_i(p_s^*)$ is also not greater than R_i^{th}. This implies that user i is also a supported user under the HPC algorithm with Δ^s. Therefore, any user that can be supported by the TPC algorithm must also be supported by the HPC algorithm. Hence, (35) holds. This completes the proof that $\overline{N}_{\mathsf{HPC}} \ge \overline{N}_{\mathsf{TPC}}$.

Proof of Statement 2 in Theorem 3: For a feasible system, all users can achieve their target SINRs under the TPC algorithm, i.e., $\overline{N}_{\mathsf{TPC}} = |\mathcal{M}|$. Using the argument in the proof of the first statement of Theorem 3, we conclude that $\overline{N}_{\mathsf{HPC}}$ is also equal to $|\mathcal{M}|$. In addition, under the HPC adaptation algorithm, we have $\Gamma_i(p_{\mathsf{HPC}}^*) \ge \hat{\gamma}_i$, $\forall\, i \in \mathcal{M}$. Therefore, supposing all potential users i decrease their α_i slowly, some of them will achieve SINRs greater than the target values if $\alpha_i > 0$ at convergence. This completes the proof for the second statement of Theorem 3.

REFERENCES

[1] S. Yeh, S. Talwar, S. Lee, and H. Kim, "WiMAX femtocells: A perspective on network architecture, capacity, and coverage," *IEEE Commun. Mag.*, vol. 46, no. 10, pp. 58–65, Oct. 2008.

[2] H. Claussen, L. T. W. Ho, and L. G. Samuel, "An overview of the femtocell concept," *Bell Labs. Tech. J.*, vol. 13, no. 1, pp. 221–245, May 2008.

[3] G. de la Roche, A. Valcarce, D. Lopez-Perez, and J. Zhang, "Access control mechanisms for femtocells," *IEEE Commun. Mag.*, vol. 48, no. 1, pp. 33–39, Jan. 2010.

[4] N. Saquib, E. Hossain, L. B. Le, and D. I. Kim, "Interference management in OFDMA femtocell networks: Issues and approaches," *IEEE Commun. Mag.*, vol. 19, no. 3, pp. 86–95, Jun. 2012.

[5] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Spectrum leasing as an incentive towards uplink macrocell and femtocell cooperation," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 617–630, Apr. 2012.

[6] S. Kyuho, S. Chong, and G. D. Veciana, "Dynamic association for load balancing and interference avoidance in multi-cell networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3566–3576, Jul. 2009.

[7] A. Golaup, M. Mustapha, and L. B. Patanapongipibul, "Femtocell access control strategy in UMTS and LTE," *IEEE Commun. Mag.*, vol. 47, no. 9, pp. 117–123, Sep. 2009.

[8] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 641–646, Nov. 1993.

[9] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

[10] R. D. Yates and H. Ching-Yao, "Integrated power control and base station assignment," *IEEE Trans. Veh. Technol.*, vol. 44, no. 3, pp. 638–644, Aug. 1995.

[11] S. V. Hanly, "An algorithm for combined cell-site selection and power control to maximize cellular spread spectrum capacity," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1332–1340, Sep. 1995.

[12] F. Rashid-Farrokhi, K. J. R. Liu, and L. Tassiulas, "Downlink power control and base station assignment," *IEEE Commun. Lett.*, vol. 1, no. 4, pp. 102–104, Jul. 1997.

[13] W. J. Teng, "SINR feedback-based integrated base-station assignment, diversity, and power control for wireless networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 473–484, Jan. 2010.

[14] C. W. Sung and K. K. Leung, "A generalized framework for distributed power control in wireless networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2625–2635, Jul. 2005.

[15] K. K. Leung and C. W. Sung, "An opportunistic power control algorithm for cellular network," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 470–478, Jun. 2006.

[16] M. Rasti and A. R. Sharafat, "Distributed uplink power control with soft removal for wireless networks," *IEEE Trans. Commun.*, vol. 59, no. 3, pp. 833–843, Mar. 2011.

[17] M. Rasti, A. R. Sharafat, and J. Zander, "Pareto and energy-efficient distributed power control with feasibility check in wireless networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 245–255, Jan. 2011.

[18] T. Alpcan, T. Basar, R. Srikant, and E. Altman, "CDMA uplink power control as a noncooperative game," *Wireless Netw.*, vol. 8, no. 6, pp. 659–670, Nov. 2002.

[19] Z. Han and K. J. R. Liu, "Noncooperative power-control game and throughput game over wireless networks," *IEEE Trans. Commun.*, vol. 53, no. 10, pp. 1625–1629, Oct. 2005.

[20] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, "Downlink power allocation for multi-class wireless systems," *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 854–867, Aug. 2005.

[21] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiminez, and L. Wynter, "A survey of network games in telecommunications," *Comput. Oper. Res.*, vol. 33, no. 2, pp. 286–311, Feb. 2006.

[22] M. Xiao, N. B. Shroff, and E. K. P. Chong, "A utility-based power control scheme in wireless cellular systems," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 210–221, Apr. 2003.

[23] H.-S. Jo, C. Mun, J. Moon, and J.-G. Yook, "Interference mitigation using uplink power control for two-tier femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 10, pp. 4906–4910, Oct. 2009.

[24] V. Chandrasekhar, J. G. Andrews, T. Muharemovic, Z. Shen, and A. Gatherer, "Power control in two-tier femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4316–4328, Aug. 2009.

[25] Y. Ji-Hoon and K. G. Shin, "Adaptive interference management of OFDMA femtocells for co-channel deployment," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 6, pp. 1225–1241, Jun. 2011.

[26] D. T. Ngo, L. B. Le, T. Le-Ngoc, E. Hossain, and D. I. Kim, "Distributed interference management in two-tier CDMA femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 979–989, Mar. 2012.

[27] S. Yanzan, R. P. Jover, and X. Wang, "Uplink interference mitigation for OFDMA femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 614–625, Feb. 2012.

[28] 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 Commun.*, vol. 28, no. 9, pp. 1479–1489, Dec. 2010.

[29] T. Alpcan, X. Fan, T. Basar, M. Arcak, and J. T. Wen, "Power control for multicell CDMA wireless networks: A team optimization approach," *Wireless Netw.*, vol. 14, no. 5, pp. 647–657, Oct. 2008.

[30] B. Steiner and P. Jung, "Optimum and suboptimum channel estimation for the uplink of CDMA mobile radio systems with joint detection," *Eur. Trans. Telecommun.*, vol. 5, no. 1, pp. 39–50, Jan./Feb. 1994.

[31] A. A. D'Amico, U. Mengali, and M. Morelli, "Channel estimation for the uplink of a DS-CDMA system," *IEEE Trans. Wireless Commun.*, vol. 2, no. 6, pp. 1132–1137, Nov. 2003.

[32] I. Shakya, F. H. Ali, and E. Stipidis, "Robust blind PIC with adaptive despreader and improved interference estimator," *Int. J. Commun. Sys.*, vol. 22, no. 1, pp. 1–23, Jan. 2009.

Vu Nguyen Ha (S'11) received the B.E. degree from the French training program for excellent engineers in Vietnam (PFIEV), Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, and an addendum degree from the École Nationale Supérieure des Télécommunications de Bretagne—Groupe des Écoles des Télécommunications, Bretagne, France, in 2007. He is currently working toward the Ph.D. degree with the Institut National de la Recherche Scientifique—Énergie, Matériaux et Télécommunications, Université du Québec, Montréal, QC, Canada.

From 2008 to 2011, he was a Research Assistant with the School of Electrical Engineering, University of Ulsan, Ulsan, Korea. His research interest includes radio resource management for wireless networks, with special emphasis on heterogeneous networks.

Long Bao Le (S'04–M'07–SM'12) received the B.Eng. degree from Ho Chi Minh City University of Technology, Vietnam, in 1999, the M.Eng. degree from the Asian Institute of Technology (AIT), Thailand, in 2002, and the Ph.D. degree from University of Manitoba, Winnipeg, MB, Canada, in 2007.

He is an Assistant Professor with the Institut National de la Recherche Scientifique—Énergie, Matériaux et Télécommunications, Université du Québec, Montréal, QC, Canada, where he leads a research group working on smart grids, cognitive radio and dynamic spectrum sharing, radio resource management, and network control and optimization.

Dr. Le is a member of the editorial board of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS and the IEEE WIRELESS COMMUNICATIONS LETTERS.