Online Optimization of Interference Coordination Parameters in Small Cell Networks

Jose A. Ayala-Romero, Juan J. Alcaraz, Javier Vales-Alonso, *Member, IEEE*, and Esteban Egea-López

***Abstract*— This paper focuses on interference coordination between the small cell and macro cell tiers of a wireless access network. We present a self-optimization mechanism for LTE-A eICIC parameters (CRE bias and ABS ratio) following a novel approach based on a model-free learning strategy, not requiring any previous knowledge about the network (e.g., topology, interference graph, and scheduling algorithms). Our proposal is built upon a stochastic optimization algorithm known as response surface methodology (RSM), that we use to find efficient eICIC configurations during network operation (online learning), adapting to changing network conditions, such as traffic or user distribution. The objective consists of optimizing a performance metric for which, in general, mathematical expression is unavailable. In particular, we consider the fifth percentile throughput defined by the 3GPP. By means of RSM, our mechanism obtains local approximations of the objective function to perform steepest ascent iterations with an adjustable level of statistical accuracy. The algorithm can be extended to account for stochastic constraints, allowing the network to optimize one performance metric while maintaining other metrics above a desired level.**

***Index Terms*— Interference coordination, heterogeneous networks, small cells, online learning.**

I. Introduction

SMALL cells are a key technology for mobile access networks, since they increase the spacial spectral reuse, enhance the network coverage, and reduce the load of the macro cells (offloading) [1]. In a typical scenario, multiple small cells overlay each macro cell, resulting in a multi-layer deployment often referred to as heterogeneous network (HetNet) [1]. With a typical frequency reuse factor of 1, interference management in HetNets is a crucial and challenging feature, especially in dense scenarios. In consequence, this issue is receiving a notable attention from both the academia and the industry. In this paper we propose a new interference management strategy based on a model-free online learning approach. One essential difference with respect to previous approaches is that our strategy is data-driven instead of model-driven, which is especially well suited for self-optimization of network parameters. Our work can be framed within the novel idea of data-driven network management described in [2]. In particular, the application scenario considered in this paper is the enhanced Inter Cell Interference Coordination (eICIC) functionality of Long Term Evolution Advanced (LTE-A) networks.

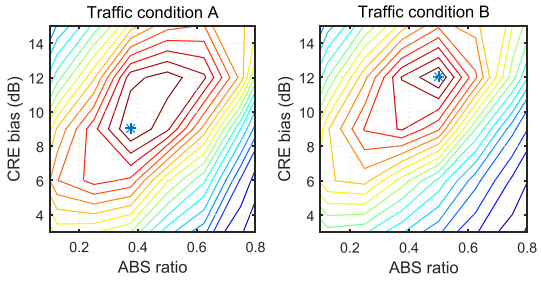


Fig. 1. 5^{th} percentile throughput (in Mbps) as a function of the eICIC parameters for the same network under two traffic conditions.

In LTE-A, the interference coordination between macro cells (macro enhanced Node B or macro eNB, in 3GPP terminology) and small cells (pico eNB) is governed by two eICIC parameters: *Cell Range Extension* (CRE) bias, which controls the proportion of User Equipments (UEs) offloaded from macro eNBs to pico eNBs; and *Almost Blank Subframe* (ABS) ratio, which indicates the proportion of subframes that the macro eNB *mutes* in its radio frame in order to reduce the interference at UEs associated to pico eNBs. The most usual limitation of previous works is neglecting the dynamic nature of the network conditions [3]–[6]. Figure 1, shows the 5^{th} percentile throughput (a usual metric for LTE-A [3], [7], [8]) obtained in a simulated scenario following the 3GPP guidelines [9]. Both surfaces were obtained for the same network layout, but using two different traffic conditions (A and B) characterized by the average traffic demand per macro sector (10 Mbps and 30 Mbps respectively) and by the fraction of users geographically close to a small cell ($2/3$ and $1/10$ respectively). The optimal (ABS ratio, CRE bias) configurations for each situation are ($3/8$, 9 dB), ($4/8$, 12 dB), respectively. This illustrates the need for dynamic reconfiguration of the eICIC parameters as traffic conditions change over time.

Previous works addressing eICIC parameter configuration rely to a large extent on mathematical models of the system, which imply simplifications such as assuming a static network situation or considering a single source of interference. Moreover, some metrics of practical interest, such as throughput

Manuscript received August 2, 2016; revised December 26, 2016 and May 13, 2017; accepted July 4, 2017. Date of publication July 21, 2017; date of current version October 9, 2017. This work was supported by project grant AEI/FEDER TEC2016-76465-C2-1-R (AIM). The work of J. A. Ayala-Romero was supported under Grant FPU14/03701. The associate editor coordinating the review of this paper and approving it for publication was M. Rossi. *(Corresponding author: Jose A. Ayala-Romero.)*

The authors are with the Department of Information and Communications Technologies, Technical University of Cartagena, 30202 Cartagena, Spain (e-mail: josea.ayala@upct.es; juan.alcaraz@upct.es; javier.vales@upct.es; esteban.egea@upct.es).

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/TWC.2017.2727483

percentiles, can be analytically intractable. To overcome these limitations, we propose a mechanism that *learns* efficient configurations by taking performance samples from the operating network itself. It is thus an online learning algorithm and, as such, it must decide at which configurations should the samples be taken at every moment, facing the inherent exploitation-exploration tradeoff. We use a Response Surface Methodology (RSM) [10] approach to address this issue, leveraging the fact that eICIC is controlled by a small number of parameters. RSM allows us to make local approximations of the objective function with statistical guarantees, and to track variations of the optimal configuration. Because it is a black-box optimization approach, it does not make any assumption on the system and does not impose additional signaling, apart from the performance samples.

Improving a fairness metric such as the 5^{th} percentile throughput, can worsen other performance metrics such as the average throughput, as shown in [11] and in the numerical results of Section V. An operator may then be interested in keeping certain performance metrics above a minimum desired level while optimizing another metric. Therefore, we show how our RSM-based mechanism can be extended to account for stochastic constraints. The rest of the paper is organized as follows. In Section II we discuss related works and summarize our contributions. The problem addressed is formulated in Section III and Section IV presents our RSM-based mechanism and its extension for comprising stochastic constraints. Numerical results and benchmark comparisons are provided and discussed in Section V.

II. Related Work and Contribution

A. Related Work

Although there is a broad literature on interference management for HetNets, our approach to the problem is novel in many aspects as we detail in this section. First, we consider the *dynamic nature of the network* as a basic element of the problem formulation, in contrast to previous works [4]–[6], [13]–[16], [20], [21] that consider a static network situation, and do not address explicitly the real-time adaptation to changing network conditions (traffic intensity, number of users, user positions).

While some previous works address the dynamic nature of the problem [7], [12], [17], [22], their approach differs from ours in some crucial aspects. The authors of [7], [17], and [22] propose heuristic mechanisms that are very scenario-dependent. For example, although [7] performs well in the usual LTE-A scenarios for which it is designed, it is not adequate for highly dense scenarios, as shown later in Sections V and IV-C. On the other hand, the proposed schemes in [7], [12], and [22] only focus on optimizing the ABS ratio while, as shown in the introduction, both eICIC parameters have a joint impact on performance [17]. Our mechanism can be extended to optimize one metric while keeping other metric above a desired performance level.

Other works like [5], [13], and [19] address the problem using reinforcement learning (RL) and [19] also considers non stationary traffic conditions. The main issue with this approach is the well known *curse of dimensionality*, i.e. the exponential growth in complexity as the state and action spaces increase. In consequence, even with approximations of the action-value function, the convergence time of RL-based algorithms is not adequate, in general, for online operation, and therefore they should be trained offline before launching them in the network (see [19]). This raises the additional difficulty of accurately reproducing the real network in a simulation model. In contrast, our proposal do not require neither offline training nor any previous knowledge about the network.

The most distinguishing feature of our proposal is that we essentially follow a *model-free* optimization approach. This contrasts with most previous works relying on *mathematical models* of the system [3], [4], [6], [12], [18]. For mathematical tractability, even the most complex models imply simplifications and assumptions that limit their application to real operating networks. For example, the authors in [6] simplify their model by considering only the strongest interference at the receivers. This allows reducing the initial NP-hard problem to a linear programming problem. Other works like [23] and [24] make use of stochastic geometry, which is an excellent tool for high-level evaluation like determining the capacity bounds of a network, but assumes that nodes are deployed according to spatial distributions like Poisson point process. In an operating network, the real network is layout is relevant for a fine-tuned parameter configuration. Other usual simplification consists of assuming round robin scheduling on the radio interface [23], which is not well suited for HetNet environments.

Finally, our strategy also differs from previous proposals in terms of *computation and signaling overhead*. For example, in [3], the algorithm computes at each iteration: the average throughput of each UE in the current iteration, the data-rate achievable by each UE if it were connected to best pico in ABS and non-ABS subframe, the data-rate achievable by each UE if it were connected to best macro, and requires information about the network topology. Similarly, the algorithms in [4] and [25] need to construct the interference graph of the whole network to operate.

We summarize the main aspects of our paper with respect to previous related works in Tables I and II. The first one addresses the characteristics of the proposed mechanism, and the second one considered the simulation models used in each work.

B. Contribution

This paper proposes a novel self-optimizing strategy to automatically adjust the interference control parameters in HetNets. In particular, our proposal is capable of finding efficient configurations for the CRE bias and ABS ratio in LTE-A eICIC under variable traffic conditions. This is done following an online learning approach. This implies that the algorithm decides, during network operation, which configuration to use at every moment using only previous performance observations. Thus, it does not require any previous knowledge about the network topology, interference graph, radio propagation conditions, and so forth. Moreover, it

TABLE I
COMPARISON OF RELATED WORKS

	[5]	[7]	[12]	[13]	[14], [15]	[3], [4]	[16]	[17]	[18]	[19]	Ours
Controlled Parameters	CRE	ABS	ABS	ABS	CRE	both	both	both	both	both	**both**
Online Operation	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes	**Yes**
Centralized/Distributed	D	both	C	D	D	C	D	D	D	D	**C**
Type of algorithm	L	H	O	L	H	O	O	H	O	L	**L**
Black box approach	No	No	No	No	No	No	No	No	No	Yes	**Yes**

(L): learning, (H): heuristic, (O): Optimization

TABLE II
SIMULATION MODEL COMPARISON IN RELATED WORKS

UEs location	Static [4], [5], [13]–[16], [20]	Dynamic [3], [7], [12], [17]–[19], **Ours**
Traffic Model	Full Buffer [3]–[5], [13]–[16], [20]	FTP [7], [12], [17]–[19], **Ours**

allows optimizing performance metrics of interest for the 3GPP and the operators, such as throughput percentiles, for which accurate analytic expressions are not available. The main novelties of our work are:

- To propose a data-driven, model-free approach for interference management in HetNets. In a broader view, we apply the novel idea of data-driven network control [2] to the particular problem of eICIC parameter configuration.
- To use Response Surface Methodology (RSM) for online learning. This technique leverages the fact that the number of controlled parameters is small, providing a cautionary exploration scheme and the capability to adapt to changing network conditions.
- To extend this approach to the case where stochastic constraints are included in the problem formulation, which allows us to take into consideration several performance metrics simultaneously.

The resulting algorithm is simple to implement and, in terms of signaling, it only requires sending the performance observations to the centralized entity where the algorithm is executed.

III. MOTIVATIONAL SCENARIO AND PROBLEM FORMULATION

A. Application Framework: eICIC in LTE-A

The eICIC functionality was introduced by the 3GPP in Release 10 (LTE-A) [26] and comprises two mechanisms for controlling the coexistence of pico and macro eNBs: CRE and ABS.

CRE allows the UEs to associate to a pico eNB when the Reference Signal Received Power (RSRP) from the pico eNB is lower than the RSRP from the macro eNB. This mechanism prevents the underuse of the radio resources at the pico eNBs due to their RSRP. To select an eNB to associate with, the UE adds the CRE bias to the pico RSRP but not to the macro RSRP, and then selects the eNB with maximum (corrected) RSRP. Thus, the higher the CRE bias, the larger the downlink footprint of the pico eNBs. However, due to the high macro eNB interference, the UEs located at the extended region (CRE region) experience a poor Signal to Interference and Noise Ratio (SINR).

Hence, the introduction of **ABS** is motivated by the need to improve the performance of UEs located at the CRE regions. This eICIC mechanism allows the macro eNBs to mute all the data symbols in some subframes, referred to as *Almost Blank Subframes*. In these protected subframes, the SINR of downlink pico transmission is notably improved because the macro interference is removed. The protected subframes are inserted following a periodic pattern lasting 8 subframes. Therefore, it is necessary to configure the ratio of ABS subframes over conventional subframes $(0/8, \ldots, 8/8)$ within the ABS pattern. The system can also select randomly between two ABS ratios (e.g. $3/8$, $4/8$ with probability 0.5 each) resulting in an intermediate ABS ratio (e.g. $3.5/8$). Therefore in our scenario, the ABS ratio can take values from the set [0, 1]. We consider a cluster of macro eNBs sharing the same ABS pattern (synchronized muting) and the same CRE bias [8], [23], [27]. The cluster is assumed to cover a geographical area with similar traffic profile.

B. Stochastic Optimization Problem Formulation

Let us consider an operator who wants to optimize certain performance metric, referred to as F, which depends on the interference coordination parameters of the system. In eICIC, these parameters are the ABS ratio (γ) and CRE bias (ϕ). Because of the stochastic nature of the system (number and position of the terminals, traffic intensity, propagation condition of the channels, etc), $F(\gamma, \phi)$ is in general a random variable. Therefore, our goal is to find an eICIC configuration $x = (\gamma, \phi)^T$ which maximizes the expected value $E[F(x)]$.

The convex set

$$\mathcal{P} = \left\{ x = (\gamma, \phi)^T : 0 \leq \gamma \leq 1, 0 \leq \phi \leq \phi_{\max} \right\} \tag{1}$$

contains all the feasible values of x. We can formulate the main problem as

$$\max_{x \in \mathcal{P}} E[F(x)]. \tag{2}$$

We are interested in solving the above problem for a performance metric F of practical interest and for which

analytic expressions are not available. This is the case of the 5^{th} percentile throughput, proposed by the 3GPP [9] for performance evaluation of LTE scenarios under the non-full buffer traffic model, and used in multiple related works [3], [7], [8]. This metric relies on the concept of *throughput sample*, defined as the quotient between the size of a downloaded file (the initial buffer length), and the time required to download it. The 5^{th} percentile throughput is defined as the value below which 5% of the throughput samples fall.[1] This metric characterizes the performance of the worst placed UEs in terms of signal quality, and therefore maximizing it pursues max-min fairness among the UEs in the network. A hypothetical accurate model for computing $E[F(x)]$ in this case should involve the stochastic characterization of the user location, the signal to interference and noise ratio (SINR) at the users, the scheduling algorithm and the file length distribution. Such a model would be intractable and would involve unknown distributions. In general, the throughput percentiles are metrics of empirical nature.

C. Online Learning Approach

Because the objective function $E[F(x)]$ is unavailable, our proposal consists on *learning* the optimal configuration by obtaining observations of $F(x)$ from the operating network itself. This data-driven approach is known as *online learning*. Let $k = 0, 1, \ldots$ enumerate the samples of $F(x)$. To obtain each sample, the algorithm selects a configuration x_k and obtains an observation of the performance $y_k = F(x_k, \omega_k)$ where ω_k represents the (random) state of the system during the k-th observation period, comprising all the random variables affecting F (e.g. traffic demands, UEs positions, propagation conditions). Therefore y_k can be seen as a noisy sample of $E[F(x)]$. The configuration x_k has to be selected according to the sampling history of the system up to $k-1$, defined as the set of past configurations and observations: $\mathcal{H}_{k-1} = \{x_0, y_0, \ldots, x_{k-1}, y_{k-1}\}$. The objective, up to a given stage N, is to solve

$$\max_{x_0,\ldots,x_N \in \mathcal{P}} E\left[\sum_{k=0}^{N} F(x_k, \omega_k)\right] \tag{3}$$

Because no previous knowledge about the system is assumed, and ω_k is not directly observed, the algorithm addressing the above problem follows a *black box* optimization approach. This type of algorithm needs to *explore* the objective function, that is, to select configurations x_k whose performance is uncertain but that could provide useful knowledge to find even better configurations. Balancing exploitation (i.e. selecting the best option given the current knowledge) and exploration, is a central issue in online learning algorithms.

In our case, we face an additional challenge. Note that w_k is a stochastic process that is not stationary in general. This implies that, even if the learning algorithm eventually converges to an optimal configuration x^*, this configuration might not be optimal in future stages. Therefore, the algorithm should be able to adapt the system configuration to long term changes in user activity.

[1] In our simulated scenario the system obtains one sample of the 5^{th} percentile throughput from 1000 throughput samples

IV. Stochastic Optimization Algorithm

A. Response Surface Methodology

Let us consider an algorithm to find the optimal solution of the original problem (2) by means of a stochastic steepest-ascent search. Such type of algorithm operates iteratively, with iterations denoted by $n = 0, 1, 2, \ldots$, generating a sequence of configurations $\{x^{(n)}\} = x^{(0)}, x^{(1)}, \ldots x^{(n)}$, that, under certain conditions, converges to an optimal solution as $n \to \infty$. In particular, starting from an initial point $x^{(0)}$, the sequence $\{x^{(n)}\}$ is generated according to

$$x^{(n+1)} = \Pi_{\mathcal{P}}\left(x^{(n)} + \alpha d(x^{(n)})\right) \tag{4}$$

where $d(x^{(n)})$ is the steepest-ascent search direction at point $x^{(n)}$, α is the step-size, and $\Pi_{\mathcal{P}}$ is the projection operator over $\mathcal{P}$ (i.e. $\Pi_{\mathcal{P}}(x)$ provides the point in $\mathcal{P}$ closest to x). The projection is intended to avoid any iterate falling outside the feasible region.

Let us define $f(x) = E[F(x)]$. Because an explicit expression for f is not available, we need to use the samples of F to generate an approximate steepest ascent direction $\hat{d}(x(n))$.

A low dimensional problem as ours ($x^{(n)} \in \mathcal{P} \subset \mathbb{R}^2$), allows us to use Response Surface Methodology (RSM) [10], [28], [29] to sample the system and compute $\hat{d}(x(n))$ with an adjustable level of statistical accuracy.

In order to obtain the next iterate $x^{(n+1)}$ using (4), RSM builds a local approximation of the objective function f at the vicinity of $x^{(n)}$. The approximation consists of a linear model characterized by a parameter vector θ. For $x \in \mathbb{R}^2$, the parameter vector is $\theta = (\theta_0, \theta_1, \theta_2)^T$, and the approximation of f is given by

$$\hat{f}(x|\theta) = \theta_0 + x^T \theta_{-0} \tag{5}$$

where $\theta_{-0} = (\theta_1, \theta_2)^T$.

Therefore, RSM must compute, at each iteration n, the parameter vector $\theta^{(n)}$ for a given $x^{(n)}$. This computation requires taking samples of F in the vicinity of $x^{(n)}$. In order to explain the RSM procedure it is helpful to define first the following sets associated to iteration n:

- The set of design points $\mathcal{D}^{(n)}$: contains a finite set of configurations in the vicinity of $x^{(n)}$. These configurations should be small perturbations of $x^{(n)}$. Let $D = |\mathcal{D}^{(n)}|$ be the size of this set.
- The set of samples $\mathcal{S}^{(n)}$: contains a (possibly random) number p of configurations taken from $\mathcal{D}^{(n)}$ during iteration n, and their performance observations: $\mathcal{S}^{(n)} = \{x_k, y_k, \ldots, x_{k+p-1}, y_{k+p-1}\}$. The set $\mathcal{S}^{(n)}$ is therefore the incremental history between two consecutive steepest-ascent iterations, $\mathcal{S}^{(n)} = \mathcal{H}_{k+p} - \mathcal{H}_k$. Note that $p \geq D$, because it might be necessary to take more than D samples to generate a statistically reliable approximation of f.

Given $x^{(n)}$, the RSM mechanism computes the corresponding parameter vector $\theta^{(n)}$ according to the following procedure:

1) Generate the set $\mathcal{D}^{(n)}$ of design points in the vicinity of $x^{(n)}$.
2) Make observations of the performance F at the configurations in $\mathcal{D}^{(n)}$, and store the samples in $\mathcal{S}^{(n)}$.
3) Construct matrix $W^{(n)}$ and vector $y^{(n)}$ from the configurations and observations in $\mathcal{S}^{(n)}$ respectively, as follows

$$W^{(n)} = \begin{pmatrix} 1 & x_k^T \\ 1 & x_{k+1}^T \\ \vdots & \vdots \\ 1 & x_{k+p-1}^T \end{pmatrix}, \quad y^{(n)} = \begin{pmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+p-1} \end{pmatrix}. \quad (6)$$

4) Compute $\theta^{(n)}$ using least squares:

$$\theta^{(n)} = \left(\left(W^{(n)} \right)^T W^{(n)} \right)^{-1} \left(W^{(n)} \right)^T y^{(n)}. \quad (7)$$

Once $\theta^{(n)}$ is obtained, iteration (4) is applied with the following estimation of the steepest-ascent direction

$$\hat{d}(x^{(n)}) = \nabla \hat{f}(x^{(n)}|\theta^{(n)}) = \theta_{-0}^{(n)}. \quad (8)$$

Figure 2 illustrates the entities involved in the proposed scheme, the information exchange and the sequence of actions involved in the execution of one steepest ascent iteration. As shown in the diagram, the algorithm runs in a centralized entity, which generates the configurations $x_k, x_{k+1}, \ldots$, and gathers the observations taken form the network.

B. Sampling Strategy

Let us see how to determine, at each iteration, the set $\mathcal{D}^{(n)}$ and the number of samples p with statistical guarantees. Let $\delta_i > 0$ be the size of the perturbation on the i-th component of $x^{(n)}$. For low-dimensional vectors, we can use a *full factorial design* [28], which implies considering all possible combinations of (positive and negative) perturbations on each component of $x^{(n)}$. In particular, for $x^{(n)} \in \mathcal{P} \subset \mathbb{R}^2$, we have the following set

$$\begin{aligned} \mathcal{D}^{(n)} = \{ & x^{(n)}, x^{(n)} + (\delta_1, \delta_2)^T, \\ & x^{(n)} + (-\delta_1, \delta_2)^T, x^{(n)} + (\delta_1, -\delta_2)^T, \\ & x^{(n)} + (-\delta_1, -\delta_2)^T \} \end{aligned} \quad (9)$$

In case a design point falls outside the feasible region $\mathcal{P}$, it must be projected on $\mathcal{P}$ (the projection operator $\Pi_{\mathcal{P}}$ is omitted in (9) for the sake of clarity). The vector $\delta = (\delta_1, \delta_2)$ defines the local area where the first order approximation will be fitted. This area should be sufficiently large to show changes in the response, but not too large so as to provide highly biased $\hat{d}(x)$ estimation. We will discuss this aspect later in Section V.

If we take the same number of samples from each point in $\mathcal{D}^{(n)}$, the columns of $W^{(n)}$ will be orthogonal and will have the same number of positive and negative perturbations. This property guarantees that $\left(\left(W^{(n)} \right)^T W^{(n)} \right)$ in (7) is invertible [28].

The search direction is determined by the regression coefficients $\theta^{(n)}$ obtained from noisy samples of f. We can characterize statistically these coefficients to evaluate the accuracy of the search direction. Specifically, given $\theta^{(n)}$, we can obtain the angle φ within which the real steepest-ascent direction is contained with a high confidence degree (e.g. 0.9 or 0.95). The algorithm will take samples until φ is smaller than a desired target angle φ^*.

After obtaining $x^{(n)}$, the algorithm takes D samples (i.e. initially $p \leftarrow D$), one at each point in $\mathcal{D}^{(n)}$, builds $W^{(n)}$ and $y^{(n)}$ with these samples, and then uses (7) to obtain $\theta^{(n)}$. The variance of $\hat{f}(x|\theta^{(n)})$ is estimated by

$$\left(\hat{\sigma}^{(n)} \right)^2 = \frac{1}{p-2} \| y^{(n)} - W^{(n)} \theta^{(n)} \|^2 \quad (10)$$

and the covariance matrix of $\theta^{(n)}$ is given by

$$\hat{\chi}^{(n)} = \left(\left(W^{(n)} \right)^T W^{(n)} \right)^{-1} \left(\hat{\sigma}^{(n)} \right)^2. \quad (11)$$

Here, the j-th element on the diagonal of $\hat{\chi}^{(n)}$ is the variance of the j-th element of $\theta^{(n)}$ [30]. Let us denote by s_θ^2 the maximum variance of θ_{-0}.

In order to obtain the angle φ for a confidence degree q, we can use the following expression [10]

$$\frac{\varphi}{2} = \arcsin \left(\frac{s_\theta^2 \mathcal{F}_q(1, p-2)}{\sum_{j=1}^2 \left(\theta_j^{(n)} \right)^2} \right)^{1/2} \quad (12)$$

where $\mathcal{F}_q(1, p-2)$ is the q-th percentile of the F-distribution.

Therefore, if $\varphi > \varphi^*$, then D new samples are taken from each point in $\mathcal{D}^{(n)}$ and incorporated to $W^{(n)}$and $y^{(n)}$, p is updated by $p \leftarrow p + D$, and $\theta^{(n)}$ is recomputed with (7). The resulting RSM algorithm with confidence in the search direction is shown in Algorithm 1, and illustrated in Figure 3.

Algorithm 1 RSM With Confidence in the Search Direction

```
Select initial point x^(0) ∈ P
for n = 0, 1, 2, ... do
    Generate D^(n), set p = 0, and set φ to a value greater than φ*
    while φ > φ* do
        Obtain one sample at each configuration in D^(n) (D new samples)
        Insert the samples in W^(n) and y^(n)
        p ← p + D
        Estimate θ^(n) using (7)
        Estimate φ using (10), (11), and (12)
    end while
    x^(n+1) ← Π_P ( x^(n) + α θ_{-0}^(n) )
end for
```

The convergence of Algorithm 1 relies on the convergence of the stochastic iteration (4). There exist several results providing sufficient conditions for this iteration to converge. For example, it is possible to apply [31, Proposition 4.1] to proof the convergence of the stochastic iteration $x^{(n+1)} = x^{(n)} + \alpha^{(n)} \hat{d}(x^{(n)})$, using stepsizes that are square-summable ($\sum_{n=0}^{\infty} (\alpha^{(n)})^2 < \infty$) but not summable ($\sum_{n=0}^{\infty} \alpha^{(n)} = \infty$). One key requirement of this result is $\hat{d}(x^{(n)})$ being a sufficiently accurate estimator of $d(x^{(n)})$,

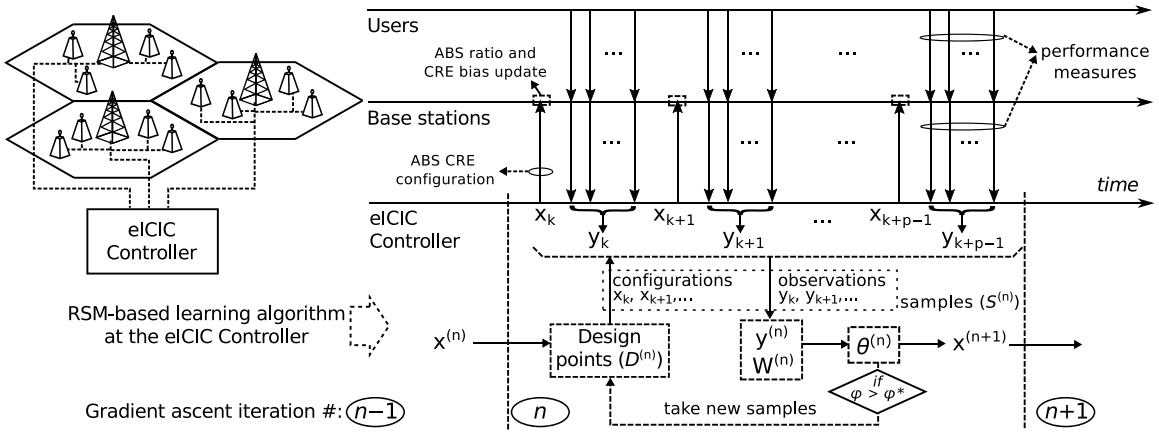


Fig. 2. Illustration of the entities participating in the proposed mechanism, the information exchange between them, and the execution of one steepest ascent iteration at the centralized entity.

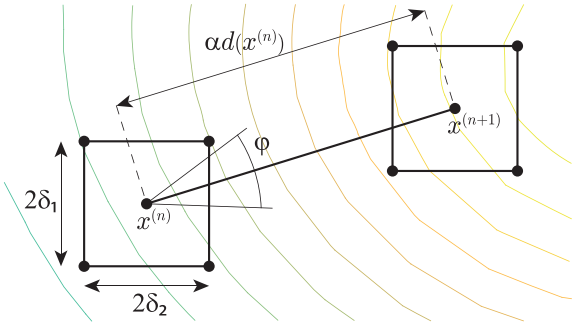


Fig. 3. Illustration of one steepest-ascent iteration of RSM with confidence in the search direction.

which is precisely what Algorithm 1 pursues with the statistical guarantees imposed on φ.

Unfortunately, this and other similar results imply assumptions on f that cannot be verified in our black-box optimization scenario. Besides, in applications requiring tracking capabilities it is convenient to use a small constant stepsize α. In this case, the convergence results state that, as n increases, $x^{(n)}$ concentrates with a high probability in a neighborhood of x^* [32]. This is the type of behavior that can be expected from Algorithm 1. Given previous reasons, it is clear that empirical experiments are required to verify this behavior in the eICIC scenario considered. Section V addresses this issue and shows numerical evaluations of the convergence rate.

C. Stochastic Optimization With Stochastic Constraints

Let us see how our mechanism can be modified to account for stochastic constraints. Let us denote the objective function by $f_0(x) = E[F_0(x)]$, and let $f_i(x) = E[F_i(x)]$, for $i = 1, \ldots, r$ be r additional performance metrics of the system. Assume that the operator wishes to maximize $f_0(x)$ while keeping all the other metrics $f_i(x)$ above a desired minimum threshold a_i for $i = 1, \ldots, r$. The functions $\{f_i\}_{i=0}^r$ cannot be directly measured. Instead, we can only obtain samples of the random responses $F_i(x)$. Therefore, our eICIC parameter configuration problem comprises now r stochastic constraints

$$\begin{aligned} \max_x \; & E[F_0(x)] \\ \text{s.t. } & E[F_i(x)] \geq a_i, \quad i = 1, \ldots, r \\ & x \geq l \\ & x \leq u \end{aligned} \tag{13}$$

where $l = (0, 0)^T$ and $u = (1, \phi_{\max})^T$, i.e. the last two constraints correspond to $x \in \mathcal{P}$. To address the stochastic optimization problem (13) we propose to modify our RSM-based algorithm by applying the following three steps:

1) *Locally approximating all the functions* $\{f_i\}_{i=0}^r$ simultaneously, using the RSM methodology and the sampling strategy described in subsection IV-B.
2) *Computing the search direction* $\tilde{d}(x^{(n)})$ for the steepest ascent iteration (4). Due to the stochastic constraints, this direction will be, in general, different to the steepest-ascent direction $d(x^{(n)})$ used in previous subsections.
3) *Obtaining the next iterate* $x^{(n+1)}$, assessing that all the constraints are satisfied.

The above procedure is based on [33], but adapted to an online learning setting. In particular, we introduce two changes: first, to compute the confidence angle for the steepest ascent direction of each function f_i; second, to use a fixed stepsize, which provides adaptability and avoids over-exploration issues, as shown in numerical evaluations.

1) Local Approximation: As in the unconstrained case, we approximate the functions f_i, $i = 0, \ldots, r$ by a first order polynomial model. With a little abuse of notation, we denote the vector of estimated coefficients for each function f_i by θ_i. Therefore, for $x \in \mathcal{P} \subset \mathbb{R}^2$ we have $\theta_i = (\theta_{i,0}, \theta_{i,1}, \theta_{i,2})^T$ for $i = 0, \ldots, r$, such that each f_i is approximated by $\hat{f}_i(x|\theta_i) = \theta_{i,0} + x^T \theta_{i,-0}$.

Following the sampling strategy explained in subsection IV-B, given a steepest ascent iterate $x^{(n)}$, we generate a set of design points $\mathcal{D}^{(n)}$. For each configuration $x_k \in \mathcal{D}^{(n)}$, we observe $r + 1$ performance metrics: $y_{i,k} = F_i(x_k, w_k)$, $i = 0, \ldots, r$, which are stored in their corresponding sets of samples $\mathcal{S}_i^{(n)}$. These sets allow us

to construct a matrix $W_i^{(n)}$ and a vector $y_i^{(n)}$ for each f_i, and then perform a least squares estimation of each coefficient vector by applying

$$\theta_i^{(n)} = \left(\left(W_i^{(n)}\right)^T W_i^{(n)}\right)^{-1} \left(W_i^{(n)}\right)^T y_i^{(n)}, i = 0, \ldots, r-1. \tag{14}$$

Once the coefficients are estimated, the problem can be locally approximated by substituting each function f_i by its linear model. To simplify the formulation, we introduce matrix $\Theta^{(n)}$ and vector $c^{(n)}$:

$$\Theta^{(n)} = \begin{pmatrix} \theta_{1,1}^{(n)} & \theta_{1,2}^{(n)} \\ \vdots & \vdots \\ \theta_{r,1}^{(n)} & \theta_{r,2}^{(n)} \end{pmatrix}, \quad c^{(n)} = \begin{pmatrix} a_1 - \theta_{1,0}^{(n)} \\ \vdots \\ a_r - \theta_{r,0}^{(n)} \end{pmatrix}. \tag{15}$$

The local approximation of (13) results in the following linear programming problem

$$\begin{aligned} &\max_{x,s,v,z} \ (\theta_{0,-0}^{(n)})^T x \\ &\text{s.t. } \Theta^{(n)} x - s = c^{(n)} \\ &\quad x + v = u \\ &\quad x - z = l \end{aligned} \tag{16}$$

where s, v and z are vectors containing the slack variables required to reformulate all the constraints as equalities. We do not need to solve the linear optimization problem (16), since $x^{(n)}$ is generated by stochastic steepest ascent iterations. Instead, we substitute x by $x^{(n)}$ in (16), to transform the constraints of (16) into a system of linear equations whose unknowns are the slack variables. The solution to this system, arranged in vectors $s^{(n)}$, $v^{(n)}$ and $z^{(n)}$, is used to derive $\tilde{d}(x^{(n)})$.

2) Computation of the Search Direction: As suggested by [28] and [33], the search direction $\tilde{d}(x^{(n)})$ can be obtained using a standard tool from interior point methods [34], known as *affine scaling method*. For our case, $\tilde{d}(x^{(n)})$ is given by

$$\begin{aligned} &\tilde{d}(x^{(n)}) \\ &= \left((\Theta^{(n)})^T (S^{(n)})^{-2} \Theta^{(n)} + (V^{(n)})^{-2} + (Z^{(n)})^{-2}\right)^{-1} \theta_{0,-0}^{(n)} \end{aligned} \tag{17}$$

where $S^{(n)}$, $V^{(n)}$ and $Z^{(n)}$ are diagonal matrices with the components of the current estimated slack vectors $s^{(n)}$, $v^{(n)}$ and $z^{(n)}$ on their respective diagonals.

Note that the inverse of the matrix within parenthesis in (17) rescales and projects $\theta_{0,-0}^{(n)}$, which corresponds to $\hat{d}(x^{(n)})$ given by (8). A geometrical intuition of how (17) modifies $\hat{d}(x^{(n)})$ is given in [28] and [33]: the modified direction $\tilde{d}(x^{(n)})$ avoids creeping along the boundary of the feasible area. The practical implication is that, for a given stepsize, the modified steepest ascent iteration approaches faster to the solution.

3) Obtaining the Steepest Ascent Iterates: Once $\tilde{d}(x^{(n)})$ is derived, the next iterate $x^{(n+1)}$ is obtained by the modified steepest ascent iteration

$$x^{(n+1)} = x^{(n)} + \alpha \tilde{d}(x^{(n)}). \tag{18}$$

Note, however, that this new iterate is not assured to lie within the feasible region. We need to incorporate a mechanism to check if all the stochastic constraints are satisfied by the new iterate. This problem was addressed in [28] and [33] using a binary search algorithm to adjust the stepsize, combined with two hypothesis tests. The first test estimates if the new iterate falls within the feasible region. The second test checks if the objective function has actually improved, and is motivated by the large stepsizes used at the start of the binary search procedure. If any test fails, the stepsize is reduced for the next stage of the binary search, and the procedure continues until the obtained iterate passes all the tests. This may imply taking a large number of low performance samples, or samples outside the feasible region, or both. This is not necessarily a problem in offline optimization, but can be a serious drawback for an operating network, since these samples may be associated to a poor quality of service/experience for the network users.

Consequently we use a constant stepsize α in (18) aiming to: 1) prevent the samples to fall too far beyond the restrictions, 2) avoid taking samples at poor performing configurations, and 3) adapt to variations on f_i for $i = 0, \ldots, r$. Once $x^{(n+1)}$ is obtained, we perform one hypotheses test at $x^{(n+1)}$ for each of the r constraints in (13). The null-hypotheses are defined as

$$H_0^{(i)} : E[F_i(x^{(n+1)})] \leq a_i \quad i = 1, \ldots, r. \tag{19}$$

If the r hypotheses $\{H_0^{(i)}\}_{i=1}^r$ are rejected, the steepest ascent estimation procedure continues at $x^{(n+1)}$; otherwise, this configuration is discarded and the iteration procedure resumes at $x^{(n)}$. This completes our proposal, referred to as the *Stepwise-Restricted RSM* (SR-RSM) algorithm, and summarized in Algorithm 2.

Algorithm 2 Stepwise-Restricted RSM

```
Select initial point x^(0) ∈ P
for n = 0, 1, 2, ... do
    Generate D^(n), set p = 0, and set φ_i > φ_i^*, for i = 0, ..., r
    while one or more φ_i > φ_i^* do
        At each x ∈ D^(n), observe r + 1 metrics {F_i}_{i=0}^r
        Insert the samples in {W_i^(n)}_{i=0}^r and {y_i^(n)}_{i=0}^r
        p ← p + D
        Estimate {θ_i^(n)}_{i=0}^r, using (14)
        Estimate {φ_i}_{i=0}^r using (10), (11), and (12)
    end while
    Obtain s^(n), v^(n), and z^(n) from (16)
    Obtain d̃(x^(n)) using (17)
    x^(n+1) ← x^(n) + α d̃(x^(n))
    Check the null-hypotheses {H_0^(i)}_{i=1}^r defined in (19)
    if one or more {H_0^(i)}_{i=1}^r are accepted then
        x^(n+1) ← x^(n)
    end if
end for
```

V. Numerical Results

A. Description of the Simulated Scenario

Let us describe the details of the simulated scenario, which is based on the 3GPP guidelines for LTE performance

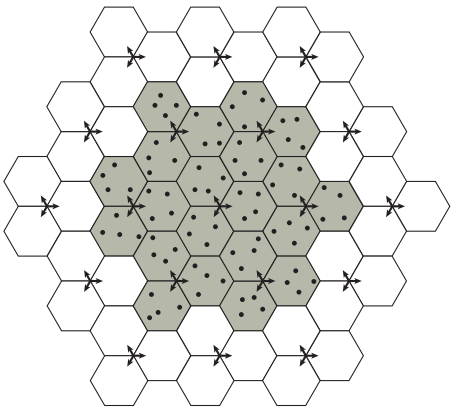


Fig. 4. Simulated topology with 4 pico eNBs per sector.

evaluation [9]. Regarding the **access network deployment and physical aspects** the scenario consists of a hexagonal grid of 19 three-sectorial macro eNBs (120 degrees per sector), that is, 57 directional macro cells. Each macro cell area contains between 2 and 12 pico cells uniformly distributed in the macro cell area and equipped with omnidirectional antennas. We simulate the 7 central macro eNBs (21 sectors) and consider the reminder (which wrap around) as an inter-cell interference source, replicating the effect of a larger network. We also consider 2x2 closed loop su-MIMO. Figure 4 illustrates the simulated topology using 4 pico eNBs per sector. The inter-site distance (ISD) is 500 m and the maximum transmit power of macro and pico cells is 46 dBm and 30 dBm, respectively. The total bandwidth of the system is 10 MHz. The wireless propagation model comprises pathloss and shadow fading [9]. Table III contains a summary of the simulation parameters.

The **traffic model** is defined by the following elements:

- Users arrive to the system according to a Poisson process. The parameter λ provides the average number of arrivals per second on each sector.
- At the user level we apply the FTP traffic model [9]. According to this model each incoming user has to download one file and remains in the system until this file is fully downloaded. The file size is 0.5 Mbytes.
- Each incoming user is dropped either at a uniformly distributed position over the macrocell area, or in a smaller area around a picocell. The first case occurs with probability P_{macro}, and the second one with $P_{\text{pico}} = 1 - P_{\text{macro}}$.

As **scheduling policy** we adopt the proposal in [35] which is an adaptation of Proportional Fair (PF) to operate in conjunction with eICIC mechanisms. This scheduler, which we will refer to as *eICIC-PF*, discriminates between different types of users. Specifically, pico UEs located at cell-edge have priority to be scheduled in ABS subframes, while pico UEs at cell-center have priority to be scheduled in non-ABS subframes.

The **SINR** is obtained by aggregating the interference of all eNBs surrounding the UE. The interference sources of each UE are: the remaining pico eNBs in the same macro sector, and the four macro eNBs causing the highest interference to the UE. In addition, we also consider the CRS interference, which is caused by the macro eNB control signals that cannot be disconnected during ABS subframes. This interference is present in approximately 10% of the resource elements of these subframes [36], [37]. Our simulations show a reduction of the achieved throughput when considering CRS interference. Nevertheless, this reduction is unevenly distributed among the UEs, and depends closely on their positions: the cell-edge UEs are the most affected since they are most likely scheduled in ABS subframes. Moreover, large values of CRE bias make the effect of CRS interference in the cell-edge UEs more relevant.

Regarding the **ABS pattern configuration**, some works like [7] and [22] consider different ABS pattern in each macro eNB (*unsynchronized muting*). With this configuration, when a macro eNBs mutes its ABS subframes according to its ABS pattern, neighboring macros with different ABS pattern can transmit in these subframes, and conversely, in non-ABS subframes the neighboring macros can mute their subframes. We adopt synchronized muting in our simulated scenario as recommended by the 3GPP [27].

B. Convergence Rate

We are interested in evaluating the influence of α, δ and φ in the convergence rate of our RSM-based learning algorithm. In this subsection we consider a reference scenario defined by Table III with 4 picos per sector, and traffic conditions given by $\lambda = 18.75$ arrivals per second (75 Mbps) per sector, and $P_{\text{pico}} = 2/3$. The experiments were repeated for scenarios with different traffic conditions and different number of small cells per sector, obtaining similar results. The metric of interest is the expected amount of samples required to achieve convergence (convergence time). This metric depends on the initial configuration $x^{(0)}$ and the criteria selected to determine convergence. Let f^* and x^* denote the optimum performance and its corresponding configuration respectively. These values were obtained in advance by extensive simulation experiments. Convergence is assumed to be attained when $|f^* - F(x_k)|$ is smaller than a given value ϵ for at least 10 consecutive samples. Different ϵ values were used. For all the convergence evaluation experiments we selected the same initial configuration $x^{(0)} \in \mathcal{P}$. Following a worst-case approach, $x^{(0)}$ is located far from x^*. In all the figures of this subsection, the convergence time for each parameter value (α, δ or φ) was obtained by averaging the results of 200 independent simulation runs.

Figure 5 depicts the convergence time versus α, showing an interesting relation between accuracy and convergence time. On the one hand, smaller values of α allow the system to operate closer to the optimum performance f^*, but requiring more samples to attain convergence. Note, however, that increasing α beyond certain value, also increases the convergence time, because a too large stepsize causes oscillations around the optimal configuration x^*.

Figure 6 shows the convergence time as a function of $\|\delta\|$. It is known that larger values of $\|\delta\|$ imply larger bias in the estimation of the steepest ascent direction. As the accuracy of $\hat{d}(x)$ decreases, the algorithm requires more samples to

TABLE III
SIMULATION PARAMETERS

Network layout	19 macro eNBs, 57 directional macro cells, 500 m ISD, [2 ... 12] pico cell per sector
System Bandwidth	10 MHz
Frame duration	Subframe 1 ms, Protected-subframe pattern 8 ms, Frame 10 ms
Transmit power	Macro cell 46 dBm, pico cell 30 dBm
Antenna Pattern (macro sector)	$A_H(\phi) = -min\left[12\left(\frac{\phi}{\phi_{3dB}}\right)^2, A_m\right]$, $A_m = 70$ degrees $A_m = 25$ dB
Antenna Pattern (pico)	Omnidirectional
Antenna gains	macro: 14 dBi; pico: 5 dBi
Macro to UE path loss	128.1 + 37.6 · $log_{10}(R[\text{Km}])$ where R is the macro eNB to UE distance
Pico to UE path loss	149.7 + 36.7 · $log_{10}(R[\text{Km}])$ where R is the pico cell to UE distance
Shadow fading	Lognormal distribution with 10 dB standard deviation
Thermal Noise	-176 dBm
Scheduling Algorithm	eICIC Proportional Fair (eICIC-PF) [35]
Traffic Model	File Transfer Protocol (FTP)
File size	0.5 Mbytes
λ [UEs/s] (Offered traffic load) [Mbps])	0.5,...,32.5 UEs/s (2,...,130 Mbps)
Minimum distances	Macro - pico: 70 m; Macro - UE: 35 m; Pico - pico: 40 m; Pico - UE: 10 m

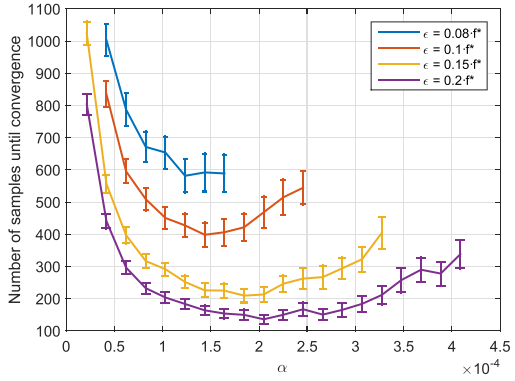


Fig. 5. Number of samples until convergence as a function of α for different ϵ values ($\delta = (0.03, 0.5)$, $\varphi^* = 7$º with a 0.95 confidence degree).

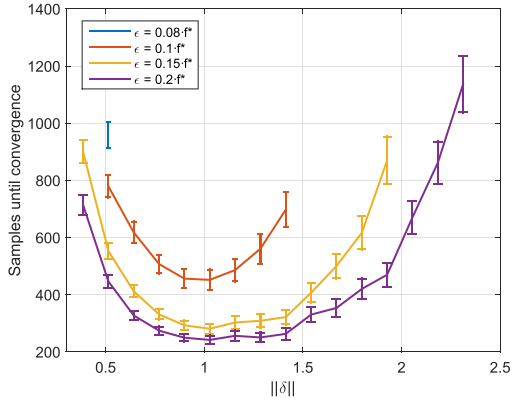


Fig. 6. Number of samples until convergence as a function of $\|\delta\|$ ($\alpha = 6.5 \cdot 10^{-5}$ and $\varphi^* = 7$º with a 0.95 confidence degree).

approach the optimum. Reducing $\|\delta\|$ below certain value also implies a slower convergence. This is because when the design points are closer to each other, their performances become more similar, and computing $\hat{d}(x^{(n)})$ requires a higher amount of samples at each iteration.

Finally, Figure 7 shows the convergence time as a function of φ^*. For this parameter, there is a clear tradeoff between

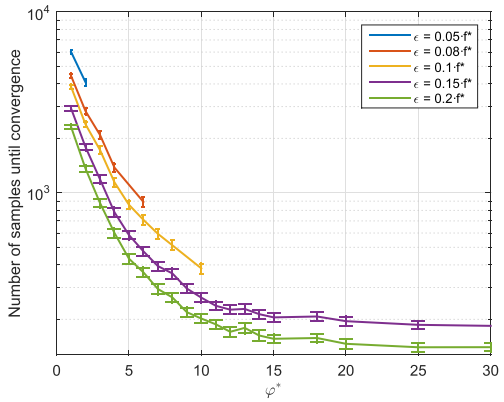


Fig. 7. Number of samples until convergence as a function of φ^* ($\alpha = 6.5 \cdot 10^{-5}$, $\delta = (0.03, 0.5)$).

accuracy and convergence rate. Increasing φ^* implies faster convergence since fewer samples are required to estimate the search direction, but smaller φ^* values are capable of operating closer to the optimum. Based on these results, the reference configuration selected for the following sections is $\alpha = 6.5 \times 10^{-5}$, $\delta = (0.03, 0.5)$, $\varphi^* = 7$º with a 0.95 confidence degree.

C. Time-Domain Performance

Let us first show how the selected configuration operates for the *plug-and-play* scenario used in previous subsection. Figure 8 shows the eICIC parameters (γ and ϕ) generated by the algorithm over consecutive samples, and the evolution of the average performance, computed with a 60-samples sliding window. The figure also highlights a region containing 90% of the performance samples to illustrate the variance of the samples and the effects of the exploration of the algorithm. Note that these samples are the estimations of the 5^{th} percentile of the user throughput, considering all the users in the simulated network. As can be observed, in this specific simulation run, the performance approaches the optimum in approximately 500 samples, starting from an initial configuration $x^{(0)}$ which

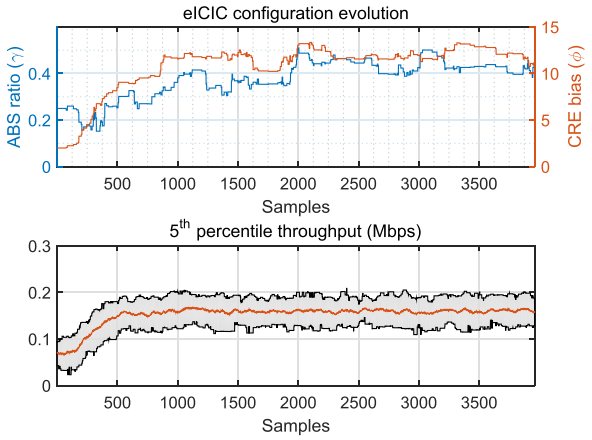


Fig. 8. Iterates $x^{(n)}$ and samples $F(x_k)$ over time.

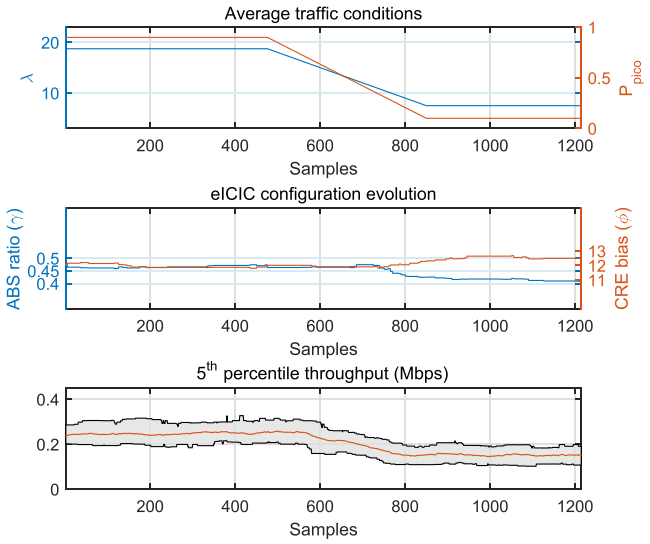


Fig. 9. Algorithm behavior under variations of the average traffic conditions.

is located far from the optimal one. It must be remarked that the simulated scenario is not static: the average traffic intensity remains constant, but the amount of active users and their locations are random processes.

We are especially interested in the ability of the algorithm to adapt to variations on the optimum eICIC configuration x^*, when the average traffic conditions change over time. For this purpose, we make the network traffic change from $\lambda = 18.75$ arrivals per second (75 Mbps) per sector, $P_{\text{pico}} = 9/10$ to $\lambda = 7.5$ arrivals per second (30 Mbps) per sector, $P_{\text{pico}} = 1/10$ (see Figure 9), over a period of approximately 400 samples. For these traffic conditions we have obtained, in advance, eICIC configurations performing very close to the optimum: $(\gamma, \phi) = (12, 0.45)$ for the initial state, and $(\gamma, \phi) = (12.5, 0.4)$ for the final state. In this figure we see the change forced on the average traffic conditions, the eICIC configuration generated by the algorithm at each sample, and the corresponding performance values. Note that the adaptation occurs in a period of time shorter than the convergence time shown in previous figures.

D. Benchmark Evaluation

We consider the following three benchmarks:

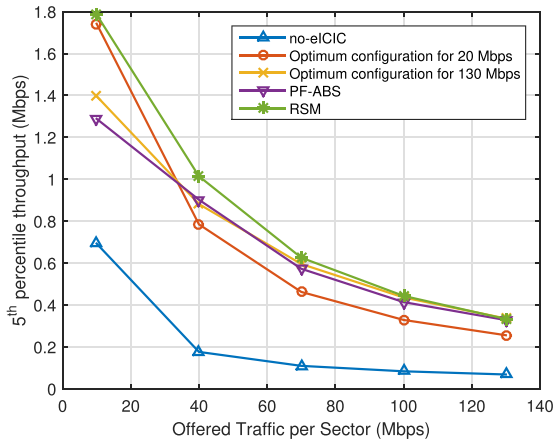
- **No eICIC**. The eICIC functionality is simply switched off in the network. This benchmark is intended to illustrate the necessity of this feature and to provide a baseline.
- **Optimal configurations for fixed traffic conditions**. We consider the optimal configuration at 4 specific traffic conditions: (20 Mbps, $P_{pico} = 2/3$), (130 Mbps, $P_{pico} = 2/3$), (20 Mbps, $P_{pico} = 7/8$), and (20 Mbps, $P_{pico} = 7/8$). These values were obtained by an exhaustive search over a set of discretized values of $\mathcal{P}$. They provide an estimate of the best achievable performance in these particular conditions.
- **Proportional-Fair-Based ABS (PF-ABS)**. This state-of-the-art heuristic proposed in [7], modifies γ at each frame according to the average of the Proportional Fair indexes. This average is used as an indicator of whether the performance is improving or deteriorating, allowing PF-ABS to decide upon the resources (subframes) allocated to the macro eNBs and the pico eNBs. As a result, different ABS patterns emerge among the macro eNBs (unsynchronized muting). In contrast to our mechanism, PF-ABS does not include CRE bias control. In the original proposal this task is left to another algorithm [38], for which we assume optimal operation.

Figure 10 shows the average value of the 5^{th} percentile throughput attained by each mechanism. In this figure, the traffic intensities per macro sector range from 10 to 130 Mbps and each sector contains 10 small cells. Two traffic distributions are considered: $P_{\text{pico}} = 2/3$ in Figure 10 (a) and $P_{\text{pico}} = 7/8$ in Figure 10 (b).

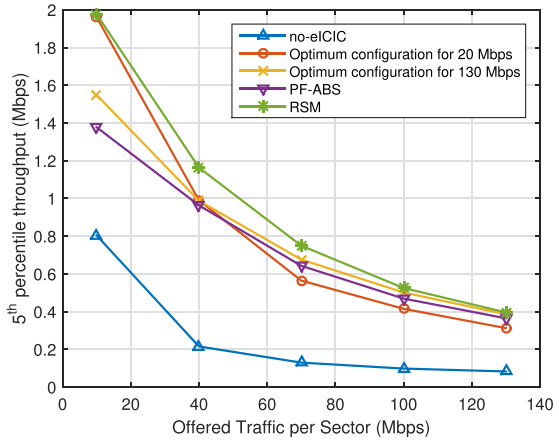
E. Evaluation of SR-RSM

To evaluate the algorithm with stochastic constraints (SR-RSM), we consider two types of benchmarks: first, we use a set of similar RSM-based algorithms allowing us to assess the effectiveness of the main design decisions of our mechanism. Second, in order to evaluate the achievable performance with respect a state-of-the-art eICIC scheme, we consider the PF-ABS scheme [7] again. The algorithms of the first type are:

- **Line Search RSM (LS-RSM)** [28] is, as previously explained, the base of our proposal. The difference is that LS-RSM adjusts the stepsize at each iteration by a binary search algorithm. The objective of this benchmark is to assess the convenience of our stepsize policy for an online learning setting.
- **Restriction-Checking RSM (RC-RSM)**. This algorithm is equal to SR-RSM, except that it uses the steepest ascent search direction $d(x)$ instead of $\tilde{d}(x)$. The objective is to quantify the effect of using $\tilde{d}(x)$ in terms of convergence rate. For RC-RSM we use two α values: $\alpha = 6.5 \cdot 10^{-5}$ (same as RSM), and $\alpha/2$. The first one aims to accelerate convergence, and the second one to operate within the feasible region.
- **Algorithm 1 (RSM)**, provides a baseline performance when not considering stochastic constraints.



(a) $P_{\text{pico}} = 2/3$, $P_{\text{macro}} = 1/3$ and 10 pico eNBs per sector



(b) $P_{\text{pico}} = 7/8$, $P_{\text{macro}} = 1/8$ and 10 pico eNBs per sector

Fig. 10. 5^{th} percentile throughput as a function of offered traffic load for the proposed benchmarks.

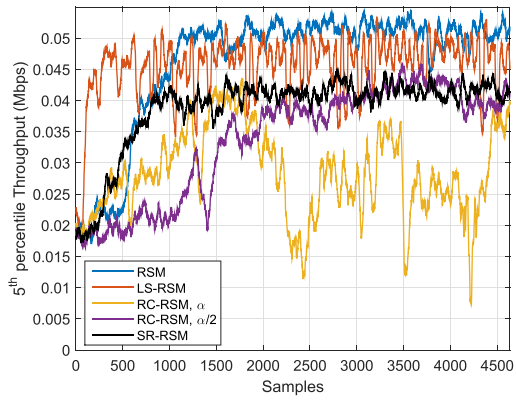


Fig. 11. Samples of the 5^{th} percentile throughput in the simulated network, obtained by each algorithm.

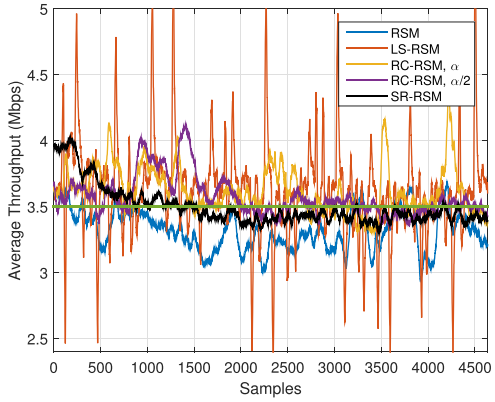


Fig. 12. Samples of the average per-user throughput in the simulated network, obtained by each algorithm.

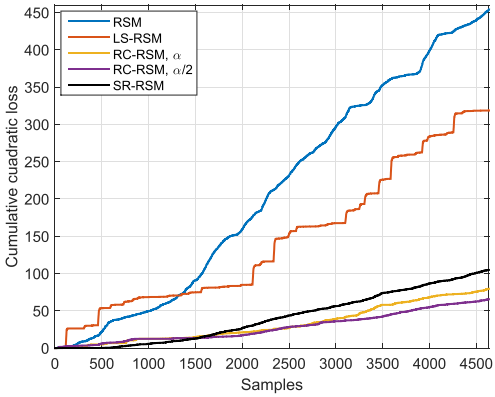


Fig. 13. Cumulative quadratic loss over time.

We use a time-domain evaluation approach to compare with other RSM-based benchmarks using a *plug-and-play* scenario in which all the algorithms start at the same initial configuration $x^{(0)}$ located far from the optimum. We consider 4 pico eNBs per sector, $\lambda = 18.75$ arrivals per second (75 Mbps) per sector and $P_{\text{macro}} = 0.9$. The objective performance metric (f_0) is the 5^{th} percentile throughput. Our simulations show that, in some scenarios, when this metric is optimized, the average user throughput (f_1) decreases. Assuming that the operator desires to keep f_1 above a desired value (a_1), we introduce the stochastic constraint $E[F_1(x)] = f_1(x) \geq a_1$.

Figure 11 shows the samples of F_0 obtained by each algorithm, and Figure 12 shows the corresponding F_1 samples. Note that RSM attains the highest F_0 values in Figure 11, but its F_1 values lie clearly below the constraint boundary. This suggests that the optimum of the constrained problem is located at the restriction boundary $f_1(x) = a_1$. These figures show why, although LS-RSM approaches very fast to the neighborhood of the optimum, it is not adequate for an online setting: it presents large fluctuations in both metrics because it takes samples at very long distances from the current configuration $x^{(n)}$. The faster convergence of SR-RSM compared to RC-RSM illustrates the convenience of the modified search direction $\tilde{d}(x)$ instead of $\hat{d}(x)$. Increasing the stepsize accelerates the convergence of RC-RSM but can cause instability as shown in Figure 11.

Regarding the effectiveness in holding the constraint, Figure 13 shows the accumulated quadratic loss over consecutive samples for each algorithm (the quadratic loss per sample is defined as $(a_1 - F_1(x_k))^2$ if $F_1(x_k) < a_1$, and 0 if $F_1(x_k) \geq a_1$). The consequence of the long range exploration

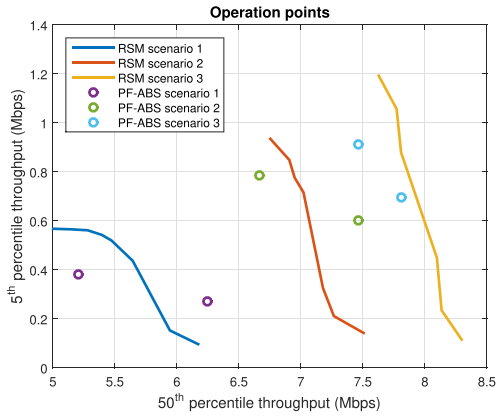


Fig. 14. SR RSM curves and PF-ABS operation points with CRE-bias configured to optimize either the 5^{th} or the 50^{th} percentile throughput. The traffic intensity in all scenarios is 100 Mbps per sector.

performed by LS-RSM is that the accumulated quadratic loss of LS-RSM is similar to the (constraint blind) RSM algorithm.

To estimate the set of performance values achievable by SR-RSM (the SR-RSM curves), we have evaluated the 5^{th} and the 50^{th} throughput percentiles by setting one metric as the objective and the other as the constraint. This was repeated in three different scenarios characterized by the number of small cells per sector (10, 14 and 18). We have also obtained, at each scenario, the performance figures for PF-ABS using two CRE-bias configurations criteria: maximizing either the 5^{th} or the 50^{th} percentile throughput. Figure 14 shows the SR-RSM curves and the PF-ABS operation points for each scenario.

Note that for 10 and 14 small cells per sector, there is one PF-ABS operation point lying on the right side of the RSM curve, implying that in these cases PF-ABS could be preferable to SR-RSM. It should be noted that PF-ABS requires unsynchronized muting, while SR-RSM is simulated under synchronized muting. Additional simulation runs have shown that when our mechanism operates with unsynchronized muting, it tends to perform better in both metrics. Moreover, Figure 14 shows that as the small cell density increases, our approach results more convenient than PF-ABS since there is a positive gap between the RSM curve and both PF-ABS configurations.

VI. Conclusion

This paper presented a novel data-based strategy for interference coordination configuration in HetNets. Specifically, it is applied to the self-optimization of LTE-A eICIC parameters: ABS ratio and CRE bias. Instead of requiring a detailed model of the system, our approach relies solely on performance measurements taken from the system. This allows us to optimize metrics of interest for the operators, such as throughput percentiles, for which analytic expressions are not available. The proposal is an online learning algorithm using RSM to obtain local approximations of the objective function to perform steepest ascent iterations. Because of the small number of parameters involved, our RSM-based scheme can provide statistical reliability on the ascent direction obtained. In addition, it is capable of adapting to variations of the objective function under non-stationary network conditions. The data-based nature of our proposal and its use of synchronized ABS imply the need of a centralized entity for its implementation. In terms of performance, numerical results show that our proposal becomes preferable to existing state-of-the-art algorithms when the deployment density of small cells increases. Our mechanism can be extended to include additional performance metrics in the form of stochastic constraints (SR-RSM), allowing the operator to optimize one metric while keeping other metrics above a desired value. In numerical evaluations, SR-RSM converged as fast as the unconstrained RSM algorithm, while keeping the performance metrics within the feasible region.

References

[1] J. Acharya, L. Gao, and S. Gaur, *Heterogeneous Networks in LTE-Advanced*. Hoboken, NJ, USA: Wiley, 2014.

[2] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big data-driven optimization for mobile networks toward 5G," *IEEE Netw.*, vol. 30, no. 1, pp. 44–51, Jan./Feb. 2016.

[3] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, "Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 137–150, Feb. 2014.

[4] A. Liu, V. K. N. Lau, L. Ruan, J. Chen, and D. Xiao, "Hierarchical radio resource optimization for heterogeneous networks with enhanced inter-cell interference coordination (eICIC)," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1684–1693, Apr. 2014.

[5] M. Simsek, M. Bennis, and I. Güvenç, "Learning based frequency- and time-domain inter-cell interference coordination in HetNets," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4589–4602, Oct. 2015.

[6] A. B. Sediq, R. Schoenen, H. Yanikomeroglu, and G. Senarath, "Optimized distributed inter-cell interference coordination (ICIC) scheme using projected subgradient and network flow optimization," *IEEE Trans. Commun.*, vol. 63, no. 1, pp. 107–124, Jan. 2015.

[7] B. Soret *et al.*, "Centralized and distributed solutions for fast muting adaptation in LTE-advanced HetNets," *IEEE Trans. Veh. Technol.*, vol. 64, no. 1, pp. 147–158, Jan. 2015.

[8] B. Soret and K. I. Pedersen, "Macro transmission power reduction for hetnet co-channel deployments," in *Proc. Global Commun. Conf. (GLOBECOM)*, Anaheim, CA, USA, Dec. 2012, pp. 4126–4130.

[9] *Evolved Universal Terrestrial Radio Access (EUTRA); Further Advancements for E-UTRA Physical Layer Aspects*, document 3GPP TR 36.814, 3rd Generation Partnership Project (3GPP), 2010.

[10] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Hoboken, NJ, USA: Wiley, 2016.

[11] A. Merwaday, S. Mukherjee, and I. Güvenç, "Capacity analysis of LTE-advanced HetNets with reduced power subframes and range expansion," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, pp. 1–19, 2014.

[12] S. Vasudevan, R. N. Pupala, and K. Sivanesan, "Dynamic eICIC—A proactive strategy for improving spectral efficiencies of heterogeneous LTE cellular networks by leveraging user mobility and traffic dynamics," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4956–4969, Oct. 2013.

[13] A. Daeinabi and K. Sandrasegaran, "A fuzzy Q-learning approach for enhanced intercell interference coordination in LTE-advanced heterogeneous networks," in *Proc. 20th Asia–Pacific Conf. Commun. (APCC)*, Oct. 2014, pp. 139–144.

[14] M. Al-Rawi, "A dynamic approach for cell range expansion in interference coordinated LTE-advanced heterogeneous networks," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Nov. 2012, pp. 533–537.

[15] S. Mishra, A. Sengupta, and C. S. R. Murthy, "Enhancing the performance of HetNets via linear regression estimation of range expansion bias," in *Proc. 19th IEEE Int. Conf. Netw. (ICON)*, Dec. 2013, pp. 1–6.

[16] N. Trabelsi, L. Roullet, and A. Feki, "A generic framework for dynamic eICIC optimization in LTE heterogeneous networks," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2014, pp. 1–6.

[17] K. I. Pedersen, B. Soret, S. B. Barcos, G. Pocovi, and H. Wang, "Dynamic enhanced intercell interference coordination for realistic networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5551–5562, Jul. 2016.

[18] M. S. Ali, P. Coucheney, and M. Coupechoux, "Load balancing in heterogeneous networks based on distributed learning in near-potential games," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 5046–5059, Jul. 2016.
[19] O.-C. Iacoboaiea, B. Sayrac, S. B. Jemaa, and P. Bianchi, "SoN coordination in heterogeneous networks: A reinforcement learning framework," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 5835–5847, Sep. 2016.
[20] D.-H. Sung and J. S. Baras, "Utility-based almost blank subframe optimization in heterogeneous cellular networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 3622–3627.
[21] W. Jin, J. Huilin, P. Zhiwen, L. Nan, Y. Xiaohu, and D. Tianle, "Joint user association and ABS proportion optimization for load balancing in HetNet," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nanjing, China, Oct. 2015, pp. 1–6.
[22] M. Al-Rawi, J. Huschke, and M. Sedra, "Dynamic protected-subframe density configuration in LTE heterogeneous networks," in *Proc. 21st Int. Conf. Comput. Commun. Netw. (ICCCN)*, Munich, Germany, Jul. 2012, pp. 1–6.
[23] M. Cierny, H. Wang, R. Wichman, Z. Ding, and C. Wijting, "On number of almost blank subframes in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 5061–5073, Oct. 2013.
[24] Y. Wang, H. Ji, and H. Zhang, "Spectrum-efficiency enhancement in small cell networks with biasing cell association and eICIC: An analytical framework," *Int. J. Commun. Syst.*, vol. 29, no. 2, pp. 362–377, 2016.
[25] L. Zhou *et al.*, "A dynamic graph-based scheduling and interference coordination approach in heterogeneous cellular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3735–3748, May 2016.
[26] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description (Release 10)*, document 36.300 v10.5.0, 3rd Generation Partnership Project (3GPP), 2011.
[27] *System Performance of Heterogeneous Networks With Range Expansion*, document 3GPP R1-100142, 3rd Generation Partnership Project (3GPP), 2010.
[28] J. P. Kleijnen, *Design and Analysis of Simulation Experiments*, vol. 20. New York, NY, USA: Springer, 2008.
[29] K. Marti, *Stochastic Optimization Methods*. Berlin, Germany: Springer, 2005.
[30] S. A. van der Geer, "Least squares estimation," in *Encyclopedia of Statistics in Behavioral Science*, Hoboken, NJ, USA: Wiley, 2005, doi: 10.1002/0470013192.bsa199.
[31] J. N. Tsitsiklis and D. P. Bertsekas, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athenea Scientific, 1996.
[32] V. S. Borkar *et al.*, *Stochastic Approximation*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[33] E. Angün, J. Kleijnen, D. den Hertog, and G. Gürkan, "Response surface methodology with stochastic constraints for expensive simulation," *J. Oper. Res. Soc.*, vol. 60, no. 6, pp. 735–746, 2009.
[34] E. R. Barnes, "A variation on karmarkar's algorithm for solving linear programming problems," *Math. Program.*, vol. 36, no. 2, pp. 174–182, 1986.
[35] K. Min and J. So, "Scheduling and positioning for the expanded region of an indoor cell in heterogeneous networks," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Busan, South Korea, Oct. 2014, pp. 685–692.
[36] *Summary of Ad Hoc Session on FeICIC Simulation Assumptions*, document 3GPP R1-112856, 3rd Generation Partnership Project (3GPP), 2011.
[37] *Performance Evaluation for FeICIC*, document 3GPP R1-113559, 3rd Generation Partnership Project (3GPP), 2011.
[38] S. Hämäläinen, H. Sanneck, and C. Sartori, Eds., *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Hoboken, NJ, USA: Wiley, 2012.

Jose A. Ayala-Romero obtained his B.Sc. degree in telematics engineering and the M.Sc. degree in telecommunication engineering from the Technical University of Cartagena (UPCT), Spain, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree in the Department of Information and Communications Technologies at UPCT. His research interests are cellular networks and learning algorithms.

Juan J. Alcaraz received the degree in telecommunications engineering from the Polytechnical University of Valencia (UPV) in 1999. After working in the telecommunications industry he joined the Technical University of Cartagena (UPCT) in 2004, where he obtained his Ph.D. in telecommunications engineering in 2007. He is currently an Associate Professor with the Department of Information and Communication Technologies at UPCT. He was a Fulbright visiting scholar at the Electrical Engineering department at UCLA in 2013. His current research focuses on learning algorithms for wireless network management.

Javier Vales-Alonso received the M.Sc. degree in telecommunication engineering from the University of Vigo, Vigo, Spain, in 2000, the Ph.D. degree in telecommunication engineering from the Technical University of Cartagena, Cartagena, Spain, in 2005, and the M.Sc. degree in mathematics from the National University of Distance Education, Madrid, Spain, in 2015. He is involved in different research topics related to modeling and optimization.

Esteban Egea-López received the degree in telecommunications engineering from the Universitat Politècnica de València, Spain, in 2002, the master's degree in electronics from the University of Gävle, Sweden, in 2001, and the Ph.D. degree in telecommunications from the Technical University of Cartagena in 2006. He is currently an Associate Professor with the Department of Information Technologies and Communications, Technical University of Cartagena. His research interest is focused on vehicular networks and MAC protocols.