Online Learning for Self-Optimization in Heterogeneous Networks

Ph.D. dissertation

José Antonio Ayala Romero
josea.ayala@upct.es

Department of Information and Communication Technologies

Supervised by:
Juan José Alcaraz Espin
juan.alcaraz@upct.es

July 2019
CONFORMIDAD DE SOLICITUD DE AUTORIZACIÓN DE DEPÓSITO DE
TESIS DOCTORAL POR EL/LA DIRECTOR/A DE LA TESIS

D. Juan José Alcaraz Espín Director de la Tesis doctoral Online Learning for Self-Optimization in Heterogeneous Networks.

INFORMA:

Que la referida Tesis Doctoral, ha sido realizada por D. José Antonio Ayala Romero, dentro del Programa de Doctorado Tecnologías de la Información y las Comunicaciones, dando mi conformidad para que sea presentada ante el Comité de Dirección de la Escuela Internacional de Doctorado para ser autorizado su depósito.

La rama de conocimiento en la que esta tesis ha sido desarrollada es:

☐ Ciencias
☐ Ciencias Sociales y Jurídicas
☒ Ingeniería y Arquitectura

En Cartagena, a 11 de Junio de 2019

EL/LA DIRECTOR/A DE LA TESIS

Fdo.: Juan José Alcaraz Espín

COMITÉ DE DIRECCIÓN ESCUELA INTERNACIONAL DE DOCTORADO
CONFORMIDAD DE DEPÓSITO DE TESIS DOCTORAL
POR LA COMISIÓN ACADÉMICA DEL PROGRAMA

D. Jorge Larrey Ruiz, Presidente/a de la Comisión Académica del Programa Tecnologías de la Información y las Comunicaciones.

INFORMA:

Que la Tesis Doctoral titulada, “Online Learning for Self-Optimization in Heterogeneous Networks”, ha sido realizada, dentro del mencionado Programa de Doctorado, por D. José Antonio Ayala Romero bajo la dirección y supervisión del Dr. Juan José Alcaraz Espín.

En reunión de la Comisión Académica, visto que en la misma se acreditan los indicios de calidad correspondientes y la autorización del Director/a de la misma, se acordó dar la conformidad, con la finalidad de que sea autorizado su depósito por el Comité de Dirección de la Escuela Internacional de Doctorado.

La Rama de conocimiento por la que esta tesis ha sido desarrollada es:

☐ Ciencias
☐ Ciencias Sociales y Jurídicas
☒ Ingeniería y Arquitectura

En Cartagena, a 12 de Junio de 2019

EL PRESIDENTE DE LA COMISIÓN ACADÉMICA

Fdo: Jorge Larrey Ruiz

COMITÉ DE DIRECCIÓN ESCUELA INTERNACIONAL DE DOCTORADO
Sr. D. José Antonio Ayala Romero

Visto el informe favorable del Director de Tesis y el Vº Bº de la Comisión Académica del Programa de Doctorado “Tecnologías de la Información y las Comunicaciones” para la presentación de la Tesis Doctoral titulada: “Online learning for self-optimization in heterogeneous networks” solicitada por D. José Antonio Ayala Romero, el Comité de Dirección de la Escuela Internacional de Doctorado de la Universidad Politécnica de Cartagena, en reunión celebrada el 25 de junio de 2019, considerando lo dispuesto en el artículo 23 del Reglamento de Estudios Oficiales de Doctorado de la UPCT, aprobado en Consejo de Gobierno el 17 de diciembre de 2015,

ACUERDA

Autorizar la presentación de la Tesis Doctoral a D. José Antonio Ayala Romero en la modalidad de “compendio de publicaciones”.

Contra el presente acuerdo, que no agota la vía administrativa, podrá formular recurso de alzada ante el Sr. Rector-Magnífico de la Universidad Politécnica de Cartagena, en el plazo de un mes a partir de la notificación de la presente.

Cartagena, 25 de junio de 2019

EL DIRECTOR DE LA ESCUELA
INTERNACIONAL DE DOCTORADO

Pd.: Pablo S. Fernández Escámez
This thesis has been elaborated as a compilation of articles:


**International mention:**

- Research stay at Dipartimento di ingegneria dell’informazione (DEI), University of Padova, Padova (Italy). Supervisor: Michele Zorzi. From September 11th, 2017 to December 10th, 2017. 3 months.

- Research stay at 5G Networks group, NEC Laboratories Europe, Heidelberg (Germany). Supervisor: Xavier Costa Perez. From September 1st, 2018 to February 28th, 2019. 6 months.
This Ph.D. thesis has been funded by:

A grant from the program “Ayudas para la Formación de Profesorado Universitario (FPU)” of the Ministry of Science, Innovation and Universities of Spain. Ref: FPU14/03701

Grants from the FPU research stay program. Ref: EST16/00520, EST17/00489.


The research stay in NEC Laboratories Europe was partially supported by them.
Agradecimientos

En primer lugar, deseo expresar mi agradecimiento al director de este trabajo, Juan José Alcaraz Espín, por su gran implicación en este proyecto durante todos estos años, por sus consejos y por ser un modelo a seguir tanto en lo profesional como en lo personal.

También me gustaría dar las gracias a Michele Zorzi, responsable del grupo SIGNET del Dipartimento di ingegneria dell’informazione en la Universidad de Padua (Italia), por invitarme a realizar una estancia en su grupo. De igual manera, agradecer a Andrea Zanella por su implicación en el trabajo realizado durante mi estancia en Padua.

Agradecer también a Xavier Costa Pérez, responsable del 5G Networks Group en NEC Laboratories Europe en Heidelberg (Alemania), por invitarme a trabajar en su grupo y por ayudarme con parte de la financiación. Igualmente, a Andrés García Saavedra por su gran implicación y su cercano seguimiento.

Quiero también agradecer a mis compañeros de laboratorio por acompañarme durante todos estos años y por hacer más ameno el día a día.

A mis amigos, por estar siempre ahí y ser una parte fundamental en el camino.

Por último, quiero dar las gracias a mis padres, por ser un apoyo incondicional en todas las decisiones que he tomado.
Abstract

The ever-increasing growth of the mobile data demands combined with the new application-specific network requirements has triggered the development of the fifth-generation (5G) of the mobile network technology. The deployment of low power base stations (small cells) is one of the key improvements in network architecture aimed at enabling 5G to meet its requirements. The network architecture resulting from the combination of legacy macro base stations and small cells is referred to as Heterogeneous Network (HetNet). This architecture presents many advantages such as frequency reuse, load balancing, and lower latency. However, HetNets also presents two main technical problems: inter-cell interference associated with the massive frequency reuse performed by the small cells, and the higher power consumption introduced by an increasing number of base stations. These problems have been addressed by means of interference coordination (IC) and energy saving (ES) mechanisms. Although the configuration of these two mechanisms has been addressed separately so far, we show in this thesis that they are highly coupled. Moreover, the configuration of IC and ES is commonly addressed using network models, which presents several limitations. In this thesis, we consider the self-optimization functionality within the Self-Organizing Networks (SON) paradigm, which is intended to address these problems by allowing the network to autonomously configure its parameters while it is operating. To implement the self-optimization functionality, we propose the use of online learning algorithms, which learn efficient network configurations from experience without explicitly knowing the accurate mathematical model of the network beforehand. The first part of this thesis addresses the configuration of the IC mechanism in HetNets. We propose several online learning model-free solutions based on different techniques such as Response Surface Method (RSM) and Multi-Armed Bandit (MAB) algorithms. We also consider stochastic constraints in the learning process. In the second part, we address the joint problem of IC and ES in HetNets proposing several solutions based on Dynamic Programming, Contextual Multi-Armed Bandit algorithms, and Machine Learning tools such as Neural Networks and Gaussian processes.
# Contents

1 Introduction ................................................. 1  
1.1 Online Learning Enabling 5G Networks ................. 1  
1.2 Heterogeneous Networks as a 5G Network Architecture ... 3  
1.3 Motivation and Objectives .............................. 4  

2 Related Work .................................................. 7  
2.1 State-of-the-art in Interference Coordination Mechanisms for HetNets 7  
2.2 State-of-the-art in Energy Saving Mechanisms for HetNets .... 11  

I Article Summary ............................................... 15  

3 Article 1: Online Learning for Interference Coordination in Heterogeneous Networks ............................................. 17  
3.1 Summary ...................................................... 17  
3.2 Methodology .................................................. 18  
3.3 Results ....................................................... 19  
3.4 Contributions ............................................... 20  
3.5 Conclusions .................................................. 22  

4 Article 2: Online Optimization of Interference Coordination Parameters in Small Cell Networks ........................................ 25  
4.1 Summary ...................................................... 25  
4.2 Methodology .................................................. 26  
4.3 Results ....................................................... 27  
4.4 Contributions ............................................... 28  
4.5 Conclusions .................................................. 30  

5 Article 3: Data-Driven Configuration of Interference Coordination Parameters in HetNets ............................................. 33  
5.1 Summary ...................................................... 33  
5.2 Methodology .................................................. 34  
5.3 Results ....................................................... 35  
5.4 Contributions ............................................... 35  
5.5 Conclusions .................................................. 37
## 6 Article 4: Energy Saving and Interference Coordination in HetNets

**Using Dynamic Programming and CEC**

6.1 Summary .................................................. 39
6.2 Methodology .............................................. 41
6.3 Results .................................................. 43
6.4 Contributions ........................................... 45
6.5 Conclusions ............................................. 46

## 7 Article 5: Contextual Bandit Approach for Energy Saving and Interference Coordination in HetNets

7.1 Summary .................................................. 47
7.2 Methodology .............................................. 48
7.3 Results .................................................. 49
7.4 Contributions ........................................... 49
7.5 Conclusions ............................................. 49

## 8 Article 6: Online Learning for Energy Saving and Interference Coordination in HetNets

8.1 Summary .................................................. 53
8.2 Methodology .............................................. 54
8.3 Results .................................................. 54
8.4 Contributions ........................................... 55
8.5 Conclusions ............................................. 56

## II Article list

### Conclusions

- Overview of the problem relevance ........................................... 129
- Summary of Contributions ............................................... 129
- Future Work .................................................. 131

### Bibliography

133
CHAPTER 1

Introduction

1.1 Online Learning Enabling 5G Networks

We are witnessing exponential growth in both connectivity and data traffic density in mobile networks which is enabled by the recent advances in the Internet of Things (IoT) domain, Machine-to-Machine (M2M) communications, cloud computing, and many other technologies. It is expected that there will be more than fifty billion devices connected to the mobile network in the next few years [1, 2, 3]. The combined effect of the growing demand and the new application-specific requirements is triggering the mobile network technology to its next major evolution, the fifth-generation (5G).

The development of the 5G technology involves the following challenges in terms of network management [4]. First, the complexity of the access network has increased due to the heterogeneity of the access nodes and ultra-dense deployments. Second, the operator does not have total control of the operation of all nodes since users can deploy their own femto cells. Moreover, energy-saving policies cause the variability of the number of active nodes in the access network, which renders a very dynamic network. Third, the network is expected to accommodate around a thousand times more traffic and ten times more users. Fourth, the network has to manage new virtualized architectures, e.g., network slicing, which allows the creation of logical network on top of a shared infrastructure enabling new business models. Fifth, new user applications require gigabyte speeds and ultra-low latency. Sixth, the operator is allowed to make use of a heterogeneous spectrum, comprising licensed and unlicensed bands.

In this challenging context, a vast number of parameters should be properly configured and tuned for an adequate operation of the network. However, this task is not straightforward since the relation between the configuration of the parameters and the network performance is usually determined by very complex and non-linear functions. One of the main difficulties is that the optimal configuration of the network parameters depends on multiple variables that vary according to stochastic processes, which can be non-stationary. This is the case, for example, of the location of the users, their traffic intensities, the number activated network...
resources (like small cells), etc. The constant variation of these variables implies that the network should adapt its configuration over time. Moreover, there is not a clear relation between the network state and the optimal configuration, which also depends on the specific network deployment. To address this challenging problem, the use of machine learning techniques is becoming a promising approach to enable network to autonomously find efficient configurations by learning from experience, which is in line with the concept of Self-organizing Networks (SONs) [4, 5].

SONs are associated with a variety of capabilities such as self-configuration to automatize the network deployment phase, self-optimization to optimize the performance of the network during its operation, and self-healing to detect and correct errors [3]. In this work, we focus on the self-optimization functionality, which allows the network to automatically configure its parameters in an online manner, i.e., while the network is operating.

Commonly, the network parameter configuration task has been performed using mathematical models of the network (model-based approach). For that, a set of parameters such as channel states or user locations are collected from the network to feed a network model. The best configuration for this network model is then used in the real network. However, this presents three main problems [7]: first, even the most complex network models imply simplifications and assumptions to make them suitable for optimization tools. This avoids many of the details and particularities of a real operating network to be captured by the network model. Second, it is necessary to assume that a set of network parameters needed to feed the network model and is fully available or perfectly known. Due to the massive number of users expected in new generation networks and the complexity of emerging radio technologies (such as Massive MIMO), the set of necessary parameters may be incomplete or inaccurate. Third, the dynamic nature of the network makes this optimization process to be continuously repeated, which may imply the increase of the signaling and computational load.

For all these reasons, we advocate for an online learning approach capable of learning efficient network configurations from experience without the need of an accurate mathematical model of the network. Moreover, it should be able to make real-time decisions in response to the dynamics of the network. Note that, in many practical scenarios, this requirement is especially challenging because of the difficulty of determining the network evolution.

The drawback of this approach is that, during the learning process, configurations associated with a poor user Quality of Service (QoS) can be selected. This is a recurrent problem in online learning algorithms, known as the exploration versus exploitation dilemma [8], where exploration means to try new configurations pursuing to find better ones but taking the risk of trying worse ones, and exploitation means to keep using the best configuration found so far.

Although online learning algorithms are intended to efficiently address the exploration vs. exploitation dilemma, we face an additional challenge: The exploration mechanism of the online algorithm should be carefully designed in order to avoid the users to experience severe QoS degradation. This makes the design of online algorithms for this context very challenging since they should avoid aggressive
1.2 Heterogeneous Networks as a 5G Network Architecture

Conventional mobile networks consider a macro cell-based planned network design, where the macro cells (referred to as macro evolved Node B or macro eNB) provide coverage to user equipments (UEs). In this type of network, all the macro eNBs have similar transmission power, antenna pattern, modulation scheme, and offer similar Quality of Service (QoS) to all UEs.

However, this type of architecture has no flexibility to address the challenges detailed above, i.e., demand growth, improved coverage, very low latency, etc. For this reason, new generation networks are adopting a more scalable and flexible approach based on a hierarchical cell architecture where low power cells (called small cells) are deployed overlapping the macro eNB coverage area. The resulting network architecture is referred to as heterogeneous network (HetNet) [9].

This architecture presents several advantages. It increases the overall network capacity leveraging the spatial reutilization of the radio spectrum performed by the small cells. It reduces the latency and offloads the macro UEs to the small cells. HetNets can also adapt to the intensity and spatial distribution of the user traffic by switching on or switching off small cells depending on the user traffic demands. Furthermore, additional small cells can be deployed ad-hoc for special events (concerts, sports games, etc.).

However, HetNets also present some technical problems. The first one is the inter-cell interference. Small cells are deployed overlapping the macro eNB coverage area and using the same bandwidth. This implies severe interference for users associated with a small cell and located at the edge of its coverage area since these users are receiving high interference power from the macro cell. This also limits the footprint of the small cell. The larger the footprint of the small cell, the higher the interference of the users located at the edge of the coverage area. This effect can lead to an underuse of small cell resources. In this work, we consider the enhanced Inter-Cell Interference Coordination (eICIC) as an interference coordination mechanism [10]. The two main features of eICIC are the Cell Range Expansion (CRE) bias which regulates the offloading from macro to small cells, and the Almost Blank Subframe (ABS) ratio which distributes the radio resources between the macro and small cells in the time domain.

Another problem present in HetNets is the increase in the power consumption associated with the increase in the number of cells. Previous studies support this statement arguing that base stations are responsible for the 80% of the power consumption in the network [11, 12]. Having said that, switching on/off the small cells seems to be a good strategy. Nevertheless, the use of this strategy is not straightforward since it is difficult to predict its effect on the users’ QoS. The reason is that the users that were associated with the small cells that are switched off need to re-associate with another cell, which can be the macro cell or another
small cell. The selection of the new cell depends basically on the Reference Signal Received Power (RSRP) which mainly depends on the network deployment, the channel quality at this moment. Moreover, the channel quality of these users will generally worsen with this change. Therefore, an inappropriate switching off strategy can lead to the saturation of the base station that was initially working correctly, with the associated degradation of the user’s QoS.

The switching on/off strategy directly affects the interference profile in the network since the number of active cells renders the interference at every user receiver. On the other hand, some works have shown that the proper configuration of the interference coordination mechanism can improve the network performance in terms of QoS [13, 14, 15]. When the network performance increases, fewer resources are needed to satisfy the demand, and therefore, fewer cells are required to be activated. Based on that, it is clear that these two problems are highly coupled. When considering a network with an energy-saving mechanism, we should also consider the configuration of the interference coordination mechanism. However, these two problems have been studied separately so far.

1.3 Motivation and Objectives

We have seen in the previous section the importance of the heterogeneous network architecture in new generation networks such as 5G. Two of the most relevant problems of this type of architecture are the inter-cell interference and energy consumption. The self-optimization functionality within the SON paradigm is intended to address these problems by allowing the network to autonomously configure its parameters while it is operating. In this work, we address the design of online learning algorithms to implement self-optimization functionalities. We formulate the following general objectives:

1. To address the Interference Coordination problem in HetNets using online learning (Articles 1, 2, and 3).
2. To address the Energy Saving problem in HetNets using online learning. Combine this with the Interference Coordination problem to find a joint solution (Articles 4, 5, and 6).

In order to accomplish the above general objectives, we formulate the following specific objectives:

1. To address the Interference Coordination problem using online model-free stochastic optimization. Consider the constrained version of the problem taking into account several metrics as constraints. Related to general objective 1 and addressed in articles 1 and 2.
2. To address the Interference Coordination problem using a Multi-armed bandit based algorithm. Related to general objective 1 and addressed in article 3.
3. To address the joint problem of Energy Saving and Interference Coordination using Approximate Dynamic Programming. Evaluate the gains with respect
to addressing Energy Saving and Interference Coordination separately. Related to general objective 2 and addressed in article 4.

4. To address the joint problem of Energy Saving and Interference Coordination as a Contextual Multi-armed bandit problem. Related to general objective 2 and addressed in articles 5 and 6.
2.1 State-of-the-art in Interference Coordination
Mechanisms for HetNets

As stated in the previous chapter, HetNets are envisioned to be a key technology in 5G networks, allowing frequency reuse, load balancing and extending the coverage to coverage holes. In this chapter we analyze two of the main problems in HetNets and explore the solutions that have been proposed until now. The first one is the inter-cell interference. So far, different mechanisms have been proposed to alleviate this problem in frequency domain (e.g., Fractional Frequency Reuse (FFR) and Carrier Aggregation (CA)) and in time domain (e.g., Coordinated Multi-Point (CoMP) and enhanced Inter-Cell Interference Coordination (eICIC)).

This work focuses on eICIC, specified in LTE-A Release 10 [10], which performs load balancing and radio resource partitioning in the time domain between macro and small cells by means of the following two mechanisms: Cell Range Expansion (CRE) bias and Almost Blank Subframe (ABS) ratio.

The CRE bias increases the footprint of the small cell by adding a bias to their Received Signal Reference Power (RSRP). It is intended to balance the offloading (from macro to pico eNBs) in the network. To select an eNB to associate with, the UE adds the CRE bias to the RSRP of the small cells but not to the macro RSRP, and then selects the base station with maximum (corrected) RSRP. Thus, the higher the CRE bias, the larger the footprint of the small cells. However, the UEs located in the extended region (CRE UEs) will generally have a poor channel quality due to the high interference received from the macro base station. Note that a CRE UE receives a stronger signal from the macro eNB than from the small cell to which it is currently associated. ABS is motivated by the need to improve the QoS performance of CRE UEs and consists of reserving certain subframes for small cell traffic only, muting data transmission from the macro eNB on some radio subframes (Almost Blank Subframes). The ABS ratio defines the portion of muted subframes over the total number of subframes (muted and not).

The optimal configuration of these parameters depends on the network state
2. Related Work

Figure 2.1: Contour lines and optimal value (marked with a * symbol) of the 5th percentile throughput in a HetNet as a function of the eICIC parameters and the number of active small cells.

in terms of UE traffic load, distribution and network topology. This is depicted in Figure 2.1, which shows the contour and the optimal point (marked with an asterisk) of the 5th percentile throughput as a function of ABS ratio and CRE bias for different network conditions. We observe in this figure that the optimal eICIC configuration changes with the traffic load and the number of active pico eNB in the network[4]. Figure 2.1 also shows that, for each network condition, the function relating the eICIC parameters and the performance follows a unimodal structure [14].

Since the network conditions change over time, it would be desirable to adjust the eICIC parameters dynamically as well. This is a challenging task since the relationship between the eICIC configuration, the network state, and its performance follows an unknown and random function and depends on each specific network deployment. We propose to learn this relationship using a model-free data-driven

1The number of active small cells in the network may change due to, for example, an energy-saving mechanism which activates or deactivates the small cells depending on the traffic load.
2.1. State-of-the-art in Interference Coordination Mechanisms for HetNets

framework [16, 17]. That is, to learn the optimal configuration from experience.

Some works like [17, 18, 19] address this problem using learning algorithms. The authors in [19] use a reinforcement learning (RL) approach, specifically a fuzzy Q-learning algorithm, to learn the optimal ABS ratio considering a fixed CRE bias. In [18], Q-learning algorithms are also proposed for ABS ratio and CRE bias learning. However, in [18, 19] the learning algorithms are applied to static network situation, i.e., constant traffic intensity and UEs with fixed locations. In contrast, in [17], the variable nature of the network is considered, which is taken into account as a central aspect of this thesis. The main issue with RL approaches is that they require offline training before operating in the network [17]. This training is usually carried out in a network simulator introducing the additional requirement of reproducing the network in a simulator model. If the simulator model is not accurate, the RL policies may be inefficient in practice. In contrast, our approach does not need offline training or any previous knowledge about the network.

Most related works address a static network situation [20, 18, 21, 22, 23, 19, 24, 25, 26], not considering explicitly the real-time adaption to dynamic network conditions. Although some works consider the dynamic nature of the network [19, 27, 28, 29], they differ from our approach in the following aspects: First, the authors of [15, 27, 28] propose a heuristic approach which is scenario dependent. For example, in [15] they assume the use of Proportional Fair (PF) scheduling in the HetNet using the PF metric as a fundamental element of the heuristic. Second, the approaches proposed in [15, 27, 29] only configure the ABS ratio ignoring that both ABS ratio and CRE bias have a joint impact in the performance metric [28]. In contrast, our approach aims at configuring these two parameters simultaneously regardless of the particular scenario, only using data retrieved from the network.

Let us discuss the differences between model-driven schemes and our data-driven, model-free approach. Figure 2.2(a) depicts conceptually a model-driven approach followed by works like [30, 20, 22, 29, 31]. This approach needs to collect key parameters from the network (such as the location of all UEs or channel gains) to feed the model which necessarily contains assumptions and/or simplifications of the real network making it suitable for an optimization algorithm. However, the optimal configuration for the model might not necessarily be the optimal configuration of the real network due to the simplification and/or assumptions in the network model. Figure 2.2(b) shows a data-driven model-free approach that is conceptually simpler and relies on direct observations of the network performance obtained from the real network (without assumption or simplifications). This allows the decision algorithm to explore configurations aiming at finding the most efficient ones.

As an example of a model-driven approach, the scheme in [30] aims at optimizing a weighted sum of the logarithms of the user throughputs. The algorithm parameters are the interference graph of the network and the data rates that each user would obtain if attached to the nearest macro station ($r^m_u$) or to the nearest pico station ($r^p_u$). The associated signaling overhead of this scheme is higher than ours, but the main drawback is the implicit assumption of considering these data rates, ($r^m_u$, $r^p_u$) independent of the eICIC configuration. Note that this assumption is a simplification/approximation since the rate at each UE is determined by the
SINR, which, in turn, depends on the eICIC configuration, and therefore $r_u^m$ and $r_u^p$ cannot be accurately known a priori. Another example of model simplifications can be seen in [22], where the initial NP-Hard problem is reduced to a linear programming problem by considering a dominant interference environment in the problem formulation.

These assumptions, among others, are useful in setting up a convex problem but may imply that the optimal configuration for this model does not match the optimal configuration of the real system, especially if the desired performance metric is not captured by the objective function.

Other works like [32] [33] address the problem using stochastic geometry. This approach models the location of macro eNBs as a homogeneous Poisson point process (PPP). It implies that the location of each macro eNB does not depend on others. In a real network the deployment of each macro eNB is generally planed and depends on the position of the surrounding macro eNB, the terrain, attenuation profile, etc. The authors in [32] also assume a Round Robin scheduling policy, which achieves a lower performance in HetNets compared to other scheduling policies [34]. In contrast, our approach relies on real data from the operating network avoiding the simplifications introduced by mathematical models.

Finally, our approach also differs from previous works in terms of computation and signaling overhead, not imposing a noticeable burden in any of them. For example, the algorithm in [30] requires information about network topology and also computes at each iteration the following data: the data-rate achievable by each UE if it were connected to best macro, the data-rate achievable by each UE if it were connected to best picocel ABS and non-ABS subframe and the mean throughput of each UE. In a similar way, [20] and [35] require the interference graph of the network to operate.

We provide a detailed comparison between our approach and related works in Tables [2.1] and [2.2].
2.2 State-of-the-art in Energy Saving Mechanisms for HetNets

Another major issue associated with HetNets is their ever-increasing energy consumption and carbon footprint. The whole industry of information and communication technology (ICT) has been estimated to contribute to 2% of the global CO$_2$ emissions, from which the 15-20% corresponds to the mobile communication industry. The increasing awareness of the harmful impact of CO$_2$ emissions on the environment makes green networking a crucial issue for a sustainable development of current and future communication systems. [35, 37]. In particular, one of the technical challenges proposed for 5G is a 100 times energy efficiency improvement [38].

It is estimated that over 80% of the energy consumption in mobile networks is due to base station operation [11, 12]. Moreover, laboratory tests in China Mobile Communications Corporation have shown that a base station consumes about 50-60% of its maximum power consumption (achieved with maximum load) with zero traffic load. This energy consumption can be reduced to 40% if it is switched off (sleeping mode) [39]. This demonstrates that switching off the base stations is an effective way to achieve energy saving in mobile networks, especially in scenarios with low traffic load in which the remaining base stations can meet the user requirements.

Figure 2.3 shows the power consumption and the ratio of UEs satisfying a QoS objective as a function of the number of active pico eNBs and the eICIC parameters. We can observe that both mechanisms are highly coupled, i.e., the optimal eICIC configuration depends on the number active pico eNBs, and the number of pico eNBs necessary to meet a QoS objective depends on the eICIC configuration. However, these two problems have been investigated separately so far.

One of the most usual approaches in energy saving is to formulate the problem as a Markov Decision Process (MDP) [40, 41]. The inherent computational complexity of the MDPs implies the use of learning approaches such as Reinforcement Learning (RL) [41]. The complexity and scalability of a learning approach are directly related to the dimensionality of the state and control spaces (the curse of dimensionality). Other works make a compact representation of the state space using a function approximation [42, 43]. Specifically, the authors in [42, 43] propose an RL approach to determine the scheduling rules under QoS constraints. But even with this strategy, RL algorithms rely on offline learning before being implemented in a real network. This is because of the slow convergence of RL algorithms. The main drawback of this strategy is that it requires a very accurate simulation model for each specific network deployment. We address this problem following two different approaches: using approximate dynamic programming (with requires offline training) and following an online learning approach, which is conceived to operate on the real network without previous training.

Other previous works addressing interference coordination in HetNets (such as [17, 18, 14, 13]) are not applicable to this setting due to additional difficulties: the
higher dimension of the problem, the absence of the main property exploited by [13] [14] (unimodality of the system response), the presence of a second performance metric in the form of a constraint, and the inclusion of the network traffic intensity (context) in the decision making process.

The authors in [44] [45] address the problem of energy saving in HetNets exploiting the ABS configuration and show that the ABS configuration has a significant impact on power consumption. Nevertheless, eNB on-off switching is not considered.

The problem of eNB on-off switching has also been addressed as an optimization problem [46] [47] [48] [49] [50] [44] [51] [52] [53]. In these works, the optimization problem is usually addressed using iterative algorithms aimed at finding suboptimal solutions since their computational complexity is NP-Hard in most cases. Moreover, the solutions of these problems have to be recomputed whenever the network state (e.g., traffic intensity) changes. In contrast, our approach learns efficient configurations for any network state. Some of these works [47] [48] [49] [50] [51] [52] [53] also take into account the user association problem, but none of them considers interference management which, as we have shown in Figure 2.3, has a notable impact on the performance when used in combination with a energy saving mechanism.

Some works address the eNB switching problem using Stochastic Geometry [54] [55] [56] [57] [58]. Nevertheless, this approach is based on a network model with some simplifications (e.g., eNBs deployed following a Poisson Point Process, path loss as the channel model). In contrast, our approach aims at learning using real data from the network without relying on a pre-designed mathematical network model. Table 2.3 summarizes the main aspects of the most related previous works.
<table>
<thead>
<tr>
<th>Controlled Parameters</th>
<th>[18]</th>
<th>[15 27]</th>
<th>29</th>
<th>19</th>
<th>24</th>
<th>25</th>
<th>30</th>
<th>20</th>
<th>26</th>
<th>28</th>
<th>31</th>
<th>17</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Operation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Synchronized ABS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data-driven approach</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Approach</td>
<td>L</td>
<td>H</td>
<td>O</td>
<td>L</td>
<td>H</td>
<td>O</td>
<td>H</td>
<td>O</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td></td>
</tr>
</tbody>
</table>

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

Table 2.1: Comparison of interference coordination related works

<table>
<thead>
<tr>
<th>UE locations</th>
<th>Static</th>
<th>Dynamic</th>
<th>Traffic Model</th>
<th>Traffic Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[20 18 23 19 24 25 26]</td>
<td>[30 15 28 29 31 17]</td>
<td>[30 20 18 23 19 24 25 26]</td>
<td>[15 28 29 31 17]</td>
</tr>
</tbody>
</table>

Table 2.2: Simulation model comparison in interference coordination related works
Table 2.3: Comparison of energy saving related works

Part I

Article Summary
Article 1: Online Learning for Interference Coordination in Heterogeneous Networks

3.1 Summary

Heterogeneous networks are considered one of the key technologies in new generation mobile access networks since they increase the spacial spectral reuse and offload the traffic demands from the macro eNBs to pico eNBs. These features also increase the interference at the receivers, which makes the interference coordination (IC) a crucial aspect, especially in scenarios with a high density of pico eNBs. However, the configuration of the IC mechanism is not trivial for several reasons. First, the best IC configuration depends on the network state (traffic load, UE positions, number of active pico eNBs), and changes over time. Second, the problem is difficult to model since it involves a large number of interference sources with interactions all-to-all. Third, the metric of interest for the operator could not be available as a closed expression, which makes the use of classic optimization tools hard to apply.

For all these reasons, we propose a model-free approach. Using this approach we do not need any detail of the underlying model (in our case, the HetNet) which is considered as a black box, nor the metric of interest to have a closed expression form. That is, this strategy is data-driven instead of model-driven, which allows us to disassociate from the details and particularities of the underlying system.

In this chapter, we propose a model-free algorithm based on stochastic optimization referred to as Response Surface Methodology (RSM) [63, 13]. This algorithm takes performance samples from the operating network. Based on these samples, it computes a noisy estimation of the steepest-ascent direction, aiming at approaching the optimum. Once the algorithm has approached the optimum, it can track variations on the optimal configuration produced by changes in the network state.
3. Article 1: Online Learning for Interference Coordination in Heterogeneous Networks

However, the performance samples are noisy and it is not trivial to know how close the estimate of the steepest-ascent direction is to the real one. For that reason, we consider an extension of RSM that estimates the confidence in the search direction. This version of the algorithm takes samples until the confidence in the search direction meets a desired value.

Our application scenario is a HetNet with several macro eNB sectors and multiple pico eNBs overlapping the macro eNB coverage area. We consider the enhanced Inter-Cell Interference Coordination (eICIC) mechanism defined by the 3GPP for LTE-A networks. The selected objective metric is the $5^{th}$ percentile throughput defined by the 3GPP. Our numerical results show a high learning rate and responsiveness outperforming other previous proposals.

3.2 Methodology

We first made an in-deep review of the state-of-the-art in interference coordination in heterogeneous networks. We paid particular attention to work focusing on the eICIC mechanism and relying on model-free approaches in order to define our contribution. We also reviewed 3GPP reports about the eICIC mechanism and simulation guidelines for LTE-A.

We formulated the problem as a stochastic optimization problem in which the metric of interest does not have analytic expression. We also considered that the underlying function to optimize changes over time.

We formulated our solution approach referred to as the Response Surface Method algorithm. We also defined the sampling strategy in order to assure confidence in the search direction of our proposal.

We defined our application scenario. We consider a HetNet composed of a cluster of contiguous macro eNBs, and several pico eNBs overlapping the coverage area of the macro eNBs. The UEs are randomly deployed and can be associated with only one eNB (macro or pico) depending on the Reference Signal Received Power (RSRP). The HetNet implements the eICIC mechanism whose configuration parameters are the ABS ratio and the CRE bias.

We implemented our network simulation from scratch using MATLAB and following the 3GPP guidelines for LTE performance evaluation. The network consists of a hexagonal grid of 19 three-sectorial macro eNBs (120 degrees per sector), that is, 57 directional macro eNBs. Each macro eNB 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. 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 path loss and shadow fading [64]. The path loss from a macro eNB to a UE is given by $128.1 + 37.6 \cdot \log_{10}(R[\text{Km}])$ where $R$ is the macro eNB to UE distance. The path loss from a pico eNB to a UE is given by $149.7 + 36.7 \cdot \log_{10}(R[\text{Km}])$ where $R$ is the pico
eNB to UE distance. The shadow fading follows a lognormal distribution with a 10 dB standard deviation. The sectorial antenna at the eNBs follows the pattern $A_{H}(\phi) = \min[12(\frac{\phi}{\phi_{3dB}})^2, A_m]$, where $\phi_{3dB}$ 70 degrees and $A_m$ 25 dB. The 3GPP guidelines impose a minimum distance between the deployment locations of the stations. The minimum distance from a macro eNB to a pico eNB is 70 meters; from a macro eNB to a UE is 35 m; from a pico eNB to another pico eNB is 40 m; from a pico eNB to a UE 10 m.

We defined the traffic model with the following elements:

- **UEs arrive at the system according to a Poisson process.** The parameter $\lambda$ provides the average number of arrivals per second in each sector.

- **At the user level, we apply the FTP traffic model [64].** 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 macro eNB area or in a smaller area around a pico eNB.** The first case occurs with probability $P_{macro}$, and the second one with $P_{pico} = 1 - P_{macro}$.

As scheduling policy, we adopted the proposal in [34] which is an adaptation of Proportional Fair (PF) to operate in conjunction with eICIC mechanisms. This scheduler 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. We adopted synchronized muting in our simulated scenario as recommended by the 3GPP [65].

Using this network simulator, we performed three evaluations: a plug-and-play static scenario, a scenario with variations in network conditions and an evaluation for different traffic loads. For the last scenario we considered some benchmarks: i) PF-ABS, a state-of-the-art algorithm for eICIC configuration [15], ii) two static eICIC configurations which are optimal for different traffic conditions, iii) the network operating without eICIC. The performance metric of interest is the 5th percentile throughput.

### 3.3 Results

In this section, we show the main results of this work. We first evaluate a plug-and-play situation in which the learning algorithm starts with an initial eICIC configuration far from the optimum. Figure 3.1 shows the eICIC parameters ($\gamma$ and $\phi$) generated by the algorithm over consecutive steepest-ascent iterations, and the evolution of the average performance obtained with a sliding window of 60 samples. Around the average performance, we depict the region containing 90% of the
performance samples. The width of this region reflects the variance of the samples and the effects of the exploration of the algorithm.

We see that the algorithm approaches the optimum performance in 25 steepest-ascent iterations approximately. The number of samples per iteration depends directly on the configuration of the confidence in the search direction. For the configuration shown in Figure 3.1, the algorithm takes 17.5 samples on average.

To illustrate how the algorithm adapts to network variations, Figure 3.2 (top) shows a change on traffic intensity ($\lambda$) and UEs distribution ($p_{\text{pico}}$). This figure also shows how RSM adapts the configuration of the eICIC parameters (middle) and the resulting network performance (bottom).

Finally, we evaluate our proposal against the following three benchmarks:

- **No eICIC.** The eICIC functionality is switched off in the network.
- **Optimal configuration for fixed traffic conditions.** We consider the case where the operator configures its eICIC parameters assuming static traffic conditions, following, for example, a worst-case approach or assuming the most typical situation.
- **Proportional-Fair-Based ABS (PF-ABS).** This state-of-the-art heuristic proposed in [15] modifies $\gamma$ according to the average of the Proportional Fair indexes at each eNB. Given that PF-ABS does not include CRE bias control, we assume the optimal one for this evaluation.

Figure 3.3 shows the average performance for traffic intensities ranging from 10 to 130 Mbps per macro sector. Given that the other benchmark solution does not learn over time, a learning rate comparison is not possible, and we show the RSM performance after convergence. Note that the performance of our proposal is at least equal to or greater than the other algorithms. Besides, RSM shows great adaptability to any scenario.

### 3.4 Contributions

This paper proposes a novel self-optimizing strategy to adjust the interference control parameters in HetNets automatically. 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 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 [16] to the particular problem of eICIC parameter configuration.
3.4. Contributions

Figure 3.1: Evolution of the algorithm over time.

Figure 3.2: Algorithm behavior under variations of the traffic conditions.

- To use Response Surface Methodology (RSM) for online learning. This technique leverages the fact that the number of controlled parameters is small,
Figure 3.3: Evaluation of the 5th percentile throughput as a function of offered traffic load for our proposal and the proposed benchmarks.

providing a cautionary exploration scheme and the capability to adapt to changing network conditions.

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.

3.5 Conclusions

This chapter 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. In terms of performance, numerical results show that our proposal becomes preferable to existing state-of-the-art algorithms, especially when the deployment density of pico eNBs increases.

Nevertheless, there are some issues left for future research. RSM has essentially three configuration parameters. An evaluation of these parameters in terms of
convergence velocity, would be necessary. Moreover, it would be of the interest of
the operators to take into account multiple metrics. Sometimes, the optimization
of one metric implies the degradation of another one. An online learning algorithm
with constraints could handle this issue. The next chapter addresses this issue.
Article 2: Online Optimization of Interference Coordination Parameters in Small Cell Networks

4.1 Summary

In the previous chapter, we proposed a model-free algorithm based on stochastic optimization referred to as Response Surface Methodology (RSM). We showed that this proposal is suitable for the online optimization of the interference coordination mechanism in LTE-A.

In this chapter, we address some issues that were left open in the previous one. RSM has mainly three configuration parameters: i) the step-size $\alpha$, which scales the distance between consecutive iterations, ii) the target angle $\phi^*$ which configure the confidence in the estimation of the search direction, iii) the size of the perturbation $\delta$, which determines the distance of the samples to the central point of the iteration. The configuration of these parameters determines the convergence rate or the not convergence at all. In this work, we perform a thorough evaluation in order to provide reliable guidelines to configure RSM.

In addition, we face the problem in which the optimization of one metric implies the degradation of other ones that are also of the interest of the operator. To address that, we propose a novel online algorithm referred to as Stepwise-Restricted RSM (SR-RSM) [13]. This algorithm addresses the online optimization of one metric, taking into account $r$ restrictions. These restrictions maintain $r$ metrics above a threshold defined by the operator.

SR-RSM relies basically on two points. First, a new search direction taking into account the constraints, which has been derived using interior point methods [65]. This new search direction increases the convergence rate by avoiding hitting the restriction boundaries. Second, we perform $r$ hypothesis tests at each iteration intending to assess if the $r$ constraint will be fulfilled in the next step. If the $r$ hypothesis tests are rejected (the next point meets the constraints), the algorithm
continues to the next point; otherwise, the algorithm remains in the current point and repeats the iteration.

We apply SR-RSM to a scenario in which the metric to optimize is the 5th percentile throughput and the average throughput has to be maintained above a minimum value. Our simulation showed that the optimization of the 5th percentile throughput (which can be seen as a max-min criterion) implies a degradation of the average throughput.

In our numerical evaluations, we assess the efficacy of the new search direction and the suitability of SR-RSM for an online setting with respect to a previous proposal referred to as Line Search RSM (LS-RSM) \[67, 68\].

4.2 Methodology

We divide the methodology of this chapter into two parts. In the first part, we address the evaluation of the configuration parameters of RSM and, in the second part, we address the formulation, implementation, and evaluation of SR-RSM.

Regarding the evaluation of the configuration parameters of RSM, we considered the network simulator described in the previous chapter with 4 pico eNBs per sector, 75 Mbps of traffic intensity per sector and \(P_{\text{pico}} = 2/3\). Let \(f^*\) and \(F(x_k)\) denote the optimal performance and the performance achieved by the eICIC configuration \(x_k\) at stage \(k\), respectively. We define that convergence has been achieved when \(|f^* - F(x_k)| < \epsilon\) for at least 10 iterations. We swept the three configuration parameters of RSM (\(\alpha\), \(\varphi\), and \(\delta\)) measuring the number of samples required to achieve the convergence for different values of \(\epsilon\). We obtained the results by averaging 200 independent simulations runs.

Concerning the second part of the methodology, we first formulated SR-RSM. For that, we derived the new search direction, which takes into account the constraints and detailed the rest of the steps of the algorithm. We implemented SR-RSM in MATLAB using a built-in function for the hypothesis test. We evaluated three benchmarks:

- **RSM algorithm** provides a baseline performance when not considering stochastic restrictions.

- **Line Search RSM (LS-RSM)** \([67]\) is the base of our proposal. The difference is that LS-RSM adjusts the step-size at each iteration by a binary search algorithm. The objective of this benchmark is to assess the convenience of our step-size policy for an online learning setting.

- **Restriction-Checking RSM (RC-RSM)**. This algorithm is equal to SR-RSM, except that it uses the classic steepest ascent search direction (not taking into account the restrictions). The objective is to quantify the effect of using the new search direction in terms of convergence rate. For RC-RSM, we use two \(\alpha\) values: the same \(\alpha\) used by RSM, and \(\alpha/2\). The first one aims to accelerate convergence, and the second one to operate within the feasible region.
4.3. Results

We evaluated a plug-and-play scenario in which the initial point is far from the optimum. The main performance metric to optimize is the 5th percentile throughput and the restriction is set in the average throughput. The network simulator is configured with 4 pico eNBs per sector, 75 Mbps of traffic intensity per sector and $P_{pico} = 0.9$.

### 4.3 Results

As detailed in the previous section, we divide the numerical evaluation of this chapter in two parts: the evaluation of the configuration parameters of RSM, and the assessment of the SR-RSM in the constrained setting.

Figure 4.1 depicts the convergence time versus $\alpha$, showing an interesting relation between accuracy and convergence time. On the one hand, smaller values of $\alpha$ allow the system to operate closer to the optimal performance but requiring more samples to attain convergence. Note, however, that increasing $\alpha$ beyond a particular value, also increases the convergence time, because a too large step-size causes oscillations.

Figure 4.2 shows the convergence time as a function of $\|\delta\|$. It is known that larger values of $\|\delta\|$ imply a larger bias in the estimation of the steepest ascent direction. As the accuracy of the estimation of the search direction decreases, the algorithm requires more samples to approach the optimum. Reducing $\|\delta\|$ below particular 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 the search direction requires a higher amount of samples at each iteration.

Finally, Figure 4.3 shows the convergence time as a function of $\varphi^*$. For this parameter, there is a clear trade-off between accuracy and convergence rate. Increasing $\varphi^*$ implies faster convergence since fewer samples are required to estimate the search direction, but smaller $\varphi^*$ values are capable of operating closer to the optimum.

Next, we evaluate the problem with restrictions. Figure 4.4 shows the samples of the 5th percentile throughput obtained by each algorithm, and Figure 4.5 shows the corresponding average throughput samples.

Note that RSM attains the highest values of the 5th percentile throughput in Figure 4.4, but its average throughput values lie clearly below the constraint boundary. This suggests that the optimum of the constrained problem is located at the restriction boundary. 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 configuration in the current step. The faster convergence of SR-RSM compared to RC-RSM illustrates the convenience of the modified search direction. Increasing the step-size accelerates the convergence of RC-RSM but can cause instability as shown in Figure 4.4.
4. Article 2: Online Optimization of Interference Coordination Parameters in Small Cell Networks

Figure 4.1: Number of samples until convergence as a function of $\alpha$ for different $\epsilon$ values.

Figure 4.2: Number of samples until convergence as a function of $\|\delta\|$ for different $\epsilon$ values.

4.4 Contributions

In this chapter, we extend the work of the previous one and therefore, the contributions are incremental. We summarize them as follows:

- To propose a data-driven, model-free self-optimizing strategy for interference management in HetNets. In particular, we apply it to the problem of
4.4. Contributions

Figure 4.3: Number of samples until convergence as a function of $\varphi^*$ for different $\epsilon$ values.

Figure 4.4: Samples of the 5th percentile throughput in the simulated network obtained by each algorithm.

eLCIC parameter configuration. This approach does not require any previous knowledge about the network topology, interference graph, radio propagation conditions, and so forth. It allows us to optimize performance metrics of interest for the 3GPP and the operators, such as throughput percentiles, for which accurate analytic expressions are not available.
4. Article 2: Online Optimization of Interference Coordination Parameters in Small Cell Networks

![Graph showing average throughput vs. samples for different algorithms.]

*Figure 4.5: Samples of the average per-user throughput in the simulated network obtained by each algorithm*

- 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.

4.5 Conclusions

In the previous chapter, we presented a novel data-based strategy for the configuration of the interference coordination mechanism in HetNets. This chapter presents SR-RSM, which is an extension of our proposal to include additional performance metrics in the form of stochastic constraints, allowing the operator to optimize one metric while keeping other metrics above the desired value. In numerical evaluations, SR-RSM converged as fast as the unconstrained RSM algorithm, while keeping the performance metrics within the feasible region.

The adjustment of the RSM configuration parameter is critical for the algorithm in terms of performance and convergence. For that reason, we performed extensive simulation experiments for a wide range of parameter values. As a result, we obtained a set of configuration parameters that attains an excellent performance in a wide variety of HetNet scenarios. However, a finer adjustment of these parameters can increase the performance in some specific scenarios. On the other hand, depending on the variability of the network performance measurements, RSM may
require a large number of samples in order to compute the search direction with sufficient confidence. Finally, RSM continues exploring after convergence, which may imply a degradation of the performance. In the next chapter, we present a novel proposal facing these problems.
CHAPTER 5

Article 3: Data-Driven Configuration of Interference Coordination Parameters in HetNets

5.1 Summary

In previous chapters, we presented an online stochastic optimization algorithm, referred to as RSM, for the configuration of the interference coordination parameters in HetNets. Although it presented an excellent performance in a wide variety of scenarios outperforming other benchmarks, RSM presents essentially three drawbacks: first, its performance is highly dependent on the adjustment of its configuration parameters. Second, it may need a relatively large amount of sample in order to compute the search direction with sufficient confidence. And third, RSM continues exploring after convergence, which implies a performance degradation associated with the continuous sampling around the estimated optimum.

In this chapter, we proposed a novel solution to tackle these problems. Our proposal is built upon the big data-driven framework proposed in [16]. The idea behind this framework is to use data retrieved from the network to perform self-optimization tasks. From the point of view of the framework, the network is seen as a black box, which avoids the use of network models.

The framework, adapted to eICIC configuration, consists of the following data flow: we retrieve raw data from the network and process it. Then, a learning entity uses this processed data to select eICIC configurations. We propose a novel learning entity to be used into this framework, which is referred to as Data-Driven eICIC Configurator DDeC [14].

Previous works have determined that the performance metric of our interest, the 5th percentile throughput, follows a unimodal structure [13, 32]. This means that the region of the optimum configuration can be reached from any other configuration following a sequence of actions with strictly increasing performance. This characteristic has been verified in a wide range of scenarios with different configurations (traffic load, number of pico eNBs, etc.) by means of extensive simulation.
DDeC comprises three modules:

- **Action Availability Selector (AAS)** selects at each time stage the set of available controls, that is, the set of controls that can be used in the current local exploration strategy.

- **Multi-Armed Bandit with limited action availability (MAB-LAA)** selects which control to use at each time step from the set of available controls generated by the AAS.

- **Network Change Detector (NCD)** observes the performance samples from the network in order to detect variations in network conditions. When it happens, the NCD notifies the other modules.

These three modules work together in two operation modes: search and exploitation. The objective of the search period is to approach the region of the optimal configuration as fast as possible. When the algorithm reaches the optimal region, the exploitation period starts. During this operation mode, the algorithm performs a fine-tuned action exploration and avoids deviating from the optimal region. This is an advantage with respect to stochastic gradient ascent algorithms like the ones proposed in the previous chapters. DDeC has one configuration parameter which controls the number of time stages that it remains on each region. Although there is an optimal value of this parameter for each possible scenario, we have assessed that using an average value for all scenarios has a very low impact on the final performance, resulting in a relative loss of 0.0007% per stage with respect to the best configuration found in our experiments. That is, the performance of our proposal has a low dependency on the adjustment of its parameters.

We compare our proposal against two families of algorithms: Stochastic Gradient Ascent (SGA) Algorithms and Multi-Armed Bandits (MAB) Algorithms. From the set of SGA algorithm we consider One Sample Gradient (OSG) [69], Multi-Sample Gradient with 2 samples (MSG2) [70], Multi-Sample Gradient with 3 samples (MSG3) [70], and Response Surface Methodology (RSM) [71]. From the set of MAB algorithms we included $\epsilon$-greedy, $\epsilon$-greedy descent, $\epsilon$-greedy logarithmic descent, softmax [73], softmax logarithmic descent, UCB normal [74], and Thompson sampling normal [75].

We evaluate our proposal and all the benchmark algorithms under static and dynamic network conditions. Our numerical results show that DDeC outperforms all benchmark algorithms in terms of 5th percentile throughput and regret under both static and dynamic conditions. We obtained the same results using the average throughput as a performance metric demonstrating the flexibility of our proposal.

### 5.2 Methodology

We formulated the problem as an online learning problem and defined the metrics of interest, the 5th percentile throughput and the regret. Then, we formulated our proposal, DDeC, in the context of the big data-driven framework proposed in...
5.3 Results

We defined the function of each module: AAS, MAB-LAA, and NCD. We implemented our proposal using MATLAB as well as the other benchmarks (SGA and MAB algorithms). The application scenario and its implementation are detailed in the previous chapters. We evaluated the configuration parameter of DDeC to assess the impact of the variation of this parameter on the final performance. For the numerical evaluation, we considered three scenarios characterized by the density of pico eNBs. Scenarios 1, 2, and 3 have 2, 4, and 12 pico eNBs per sector, respectively.

We evaluated all the algorithms in these scenarios measuring the regret of learning and the 5th percentile throughput, which is the objective metric. Then, we repeated the same evaluation but selecting the mean throughput as the objective metric. For the dynamic scenario evaluation, we selected scenario 1 and, in the middle of the learning process, we switched to scenario 2. We assessed the response of the algorithms in terms of regret. Finally, we compared our proposal against a model-driven approach from [15].

Figure 5.3 shows the evolution of the regret over the time stages for our proposal and for the two families of algorithms, SGA and MAB, in scenario 1. We consider a plug-and-play setting and, in the case of DDeC and SGA algorithms, they are initialized in a worst-case initial configuration, far from the optimal one. Analogously, Figure 5.2 depicts the 5th percentile throughput in the same setting. Our proposal shows in both figures a better performance, achieving a close to zero slope in the regret curve. Note that the regret accumulates at each time stage the distance from the expected performance of the current configuration to the expected performance of the best one. When the slope of the regret curve is close to zero, the algorithm is very close to the optimum. That is, the higher the slope of the regret, the larger the distance between the algorithm performance and the optimal one.

Moreover, additional simulations have shown that the performance of the benchmarks (especially SGA algorithms) has a high dependency on the adjustment of its parameters. For example, if we optimize the parameters for scenario 1, the performance in scenarios 2 and 3 worsen. In contrast, DDeC has shown a similar performance in all the scenarios with the same configuration parameters.

Finally, Figure 5.3 shows the evolution of the regret under changing conditions. In this evaluation, the NCD module detects a change in the network, which allows DDeC to switch from exploitation to exploration mode. Thus, our proposal can adapt to a new network state. Our numerical evaluations show that NCD needs an average of 26 time stages to detect the change.

5.4 Contributions

The first contribution of this paper is to present a specific application of the general idea of data-driven network management, described in [16, 7], as a proof-of-
Figure 5.1: Regret evaluation in a network with stationary conditions

Figure 5.2: 5th percentile throughput in a network with stationary conditions

Figure 5.3: Regret evaluation in a network with changing conditions
concept. We show how this approach can be applied to the dynamic configuration of eICIC parameters in HetNets in our proposed mechanism: the Data-Driven eICIC Configurator (DDeC).

Our proposal finds efficient configurations of the eICIC parameters using only data retrieved from the network. This data-driven nature provides two main benefits to DDeC with respect to previous approaches: first, DDeC can operate without any previous knowledge of the underlying network (e.g., topology, scheduling algorithms); second, DDeC can optimize any measurable performance metric, even those that are analytically intractable such as the 5th percentile throughput.

DDeC incorporates the proposal of a novel online learning algorithm of the Multi-Armed Bandit (MAB) type. Compared to previous MAB algorithms, our proposal comprises the following novelties: 1) it uses a local exploration strategy which notably increases the algorithm’s efficiency, 2) it uses only a reduced set of actions on each learning/decision stage, 3) it can operate under non-stationary network conditions by means of a novel module that detects if the optimal action changes. This addresses a major challenge for online learning algorithms. In consequence, our numerical experiments show that DDeC outperforms previous proposals. Besides, it does not introduce additional computational complexity compared to previous approaches.

5.5 Conclusions

This chapter presented a Data-Driven eICIC Configurator (DDeC) for interference coordination in HetNets environments based on a big data-driven framework for network optimization. It finds efficient configurations of eICIC parameters (ABS ratio and CRE bias) operating without any previous knowledge of the network (e.g., traffic load, topology, propagation conditions) only using data retrieved from the network. This data-driven approach allows us to optimize metrics of practical interest that are analytically intractable, such as 5th percentile throughput. Our numerical results show that our proposal achieves a significant improvement with respect to the state of the art of SGA and MAB algorithms in networks with both stationary and variable conditions.

In the next chapters, we address the problem of energy saving in HetNets along with the interference coordination problem. The energy-saving problem has gained significant relevance in the last few years and will be a crucial feature in new generation networks. We will show the advantages of addressing energy-saving and interference coordination together, which had been addressed separately so far.
6.1 Summary

As we studied in previous chapters, interference coordination (IC) is a crucial aspect in heterogeneous networks (HetNet) resource management. The IC mechanism alters the interference profile in the network, which directly affects the end-user performance. On the other hand, energy efficiency has gained high relevance in mobile networks. Previous studies have shown that over 80% of the energy consumed in a mobile network is due to the base stations.

In the case of HetNets, there are two aspects enabling the energy saving (ES): first, the overlapping of the coverage areas of many eNBs that compose the HetNet, and second, the variability of the traffic demands in the different periods of the day. This provides the opportunity to switch off some eNBs, reducing the overall energy consumption. In this work, we will focus on an ES mechanism consisting of the switching on and off of the pico eNBs.

However, it is essential to highlight that the IC and ES mechanisms are strongly coupled. When some eNBs are switched off by the ES mechanism, the interference profile of the network changes and therefore, the IC mechanism has to adapt its configuration to this new situation. On the other hand, we show in previous chapters that the proper configuration of the IC mechanism increases the network performance. When the network performance increases, fewer resources are needed to satisfy the demand, and therefore, fewer eNBs are required to be activated. Having said this, it is straightforward that these two mechanisms have to be configured and optimized together. However, to the best of our knowledge, these two mechanisms had been studied separately so far.
In this chapter, we address the joint problem of energy saving and interference coordination in HetNets. We formulate the problem as a finite horizon Markov Decision Process (MDP) aiming at minimizing the overall energy consumption in the network. We also consider one constraint in the formulation: a minimum QoS requirement has to be satisfied at every time stage. It is essential to mention that the selected control (ES and IC configuration) at every time stage implies an instantaneous cost (in terms of power consumption and QoS) and will affect to future states of the MDP. Thus, the objective is to find a policy selecting a control at every time stage taking into account the instantaneous and future cost of this decision (farsighted policy). Leveraging the fact that the user traffic demands usually follow periodic daily patterns, we formulate the MDP with a finite horizon, where the horizon is limited to one day, the length of the traffic period. This reduces the computational complexity of solving the MDP significantly.

However, solving this MDP is still computationally unaffordable due to the following reasons: first, the transition to future states is random since it depends on the traffic in the future. Second, the dimensionality of the state and control spaces is very large since we are taking into account ES and IC controls and the number of ES controls grows exponentially with the number of pico eNBs. This led us to use Approximate Dynamic Programming (ADP) techniques to solve the MDP. Specifically, we apply the Certainty Equivalent Control (CEC) principle to simplify the complexity of the MDP state transitions. CEC principle applies at each time stage the control that would be optimal if the uncertain quantities (in our case, the traffic demands) were fixed at their typical values \cite{79}. Following this principle, we consider the expected values of the traffic demands in future time stages to compute the impact of the selected controls in future states. In order to forecast power consumption and QoS in future time stages, we use machine learning tools. This allows us to compute the cost-to-go function, which indicates the minimal power consumption of the network from the current stage to the end of the horizon.

Moreover, we reduce the dimensionality of the state and control spaces by dividing the decision process into two levels. At the high level, the ADP algorithm selects a global (dimensionally reduced) control, which is sent to the Control Space Augmentors (CSAs). The CSAs (one for each macro eNB) compute, from the global control, the full-dimensional control to be used in the network.

Our proposal comprises two phases: the offline phase in which the machine learning forecasters (of power consumption and QoS) are trained using network data in order to compute the estimation of the cost-to-go function; and the online phase in which, in a stage by stage fashion, a control is selected to be used in the network. The result is an ADP algorithm that selects energy-efficient farsighted configurations with QoS guarantees, and achieving near-optimal performance \cite{62}.

We evaluate our proposal in an LTE-A network simulator following the 3GPP guidelines, and the results obtained show that a joint control of the ES and IC mechanisms results in a notable performance improvement compared to a disjoint control, in terms of both energy savings and QoS guarantees. Moreover, our proposal has the advantage of being adaptable to the operator QoS requirements.
6.2 Methodology

The first step in the realization of this work was an in-depth review of the state-of-the-art in energy saving on heterogeneous networks. We paid particular attention to work relying on MDPs, dynamic programming, and reinforcement learning in order to put our contribution in context. We also examined the consumption models for base stations considered in previous research papers and 3GPP reports.

We formulated the problem as a finite horizon Markov decision process. We applied the Certainty Equivalent Control principle, which is an ADP technique reducing the computational complexity of the MDP resolution. We also divided the decision making into two levels in order to reduce the dimensionality of the state and control spaces.

We designed our proposal, an ADP algorithm, with an offline phase, in which we use machine learning tools to estimate the cost-to-go function, and an online phase, in which the time stage by time stage operation is defined.

We implemented our proposal using Python programming language and the libraries TensorFlow [77] and Scikit-Learn [78] for the machine learning tools: a neural network and a polynomial regressor, respectively.

Our application scenario is an HetNet composed of a cluster of contiguous macro eNBs, and several pico eNBs overlapping the coverage area of the macro eNBs. In this scenario, the pico eNBs are activated or in sleeping mode (on or off), depending on the energy-saving mechanism, while the macro eNB is always active. The IC functionality considered was the enhanced Inter-Cell Interference Coordination (eICIC) mechanism proposed by the 3GPP for LTE-A Networks and detailed in previous chapters. By means of eICIC parameters, ABS and CRE, we were able to configure the interference profile in the network.

The network simulation was developed in Python from scratch following the 3GPP guidelines for the evaluation of LTE networks [64]. The network layout comprises 5 sectorized macro eNBs (120 degrees) and several pico eNBs for each macro sector. The inter-site distance (distance between macro eNBs) is 500 meters. The scheduling algorithm in all eNBs is Proportional Fair. We consider only the downlink traffic. To generate the traffic load, a random number of UEs following the traffic demand pattern are randomly located in the network.

The eNB consumption model used in this work is based on the 3GPP guidelines [79] and also includes the consumption associated with the activation of a sleeping eNB [80]. The power consumption model of a pico eNB $j$ is given by:

$$C^j_{0} = e^j \cdot N_{TRX} \cdot (P_0 + R^j \cdot P_{max}) + (1 - e^j) \cdot N_{TRX} \cdot P_{sleep} + \Delta$$  \hspace{1cm} (6.1)

$$\Delta = \begin{cases} \beta \cdot P_0 & \text{when eNB } j \text{ is switched on from sleeping mode} \\ 0 & \text{otherwise} \end{cases}$$ \hspace{1cm} (6.2)

where:

- $e^j$ 1 when the pico eNB $j$ is active and $e^j$ 0 otherwise.
- $N_{TRX}$ is the number of TRXs.
• $P_{\text{max}}$ is the maximum RF output.
• $P_0$ represents the power consumption at zero RF output power.
• $R^j \in [0,1]$ is the load factor of the pico eNB $j$ and depends on the ABS ratio, the CRE bias, the traffic intensity and the location of UEs.
• $P_{\text{sleep}}$ is the power consumption of TRX components in sleep mode.
• $\beta$ is the portion of $P_0$ needed to switch on the pico eNB TRXs.

Note that $\Delta$ captures the consumption associated with switching on a sleeping pico eNB. The power consumption of the macro eNB $i$ is given by

$$C_{i}^{m} = N_{\text{TRX}} \cdot (P_{0}^{m} + R^{i} \cdot P_{\text{max}}^{m}) \cdot (1 - \gamma) + N_{\text{TRX}} \cdot P_{0}^{m} \cdot \gamma \tag{6.3}$$

where $\gamma$ denotes the ABS ratio, $P_{\text{max}}^{m}$ is the maximum power output of the macro eNB, $P_{0}^{m}$ is the power consumption at zero RF output power of the macro eNB, and $R^i$ is the load factor.

We simulated the central sector using the remaining ones to emulate the aggregated interference of a larger network. The wireless channel model comprised path loss and stochastic shadow fading. The aggregated interference at each UE receiver was calculated as the aggregation of the power received from all interfering eNBs in the sector (picos and macro) plus the interference from the macro eNBs from nearby sectors. The remaining simulation parameters are detailed in previous chapters.

Using this simulator, we, first of all, evaluated the performance of the Control Space Augmentor (CSA) strategy. The idea is to measure the losses associated with dividing the decision making into the two levels. We compared the CSA strategy against the optimal strategy, the worst-case strategy, and another strategy based on the idea behind CSAs.

We generated a dataset using the network simulator described above in which we select several configurations of ES and IC mechanism, and we measure the overall power consumption of the eNBs and the QoS of the users under these configurations. We took a total of 29000 data samples.

In order to configure the machine learning tools, the regressor, and the neural network, we divided the dataset into two parts: the training set (80% of the data) and the test set (20% of the data). We trained the polynomial regressor with different degrees and the neural network with different architectures using the training set. We evaluated the performance of the different configuration using the test set and selected the one with the highest performance. For the neural network, two hidden layers with 50 units each, ReLu activation in the hidden layers and sigmoid activation in the output layer. The degree of the polynomial regressor was set to 6. The neural network was trained using the Adam algorithm. The solver used to fit the polynomial regressor was selected automatically by Scikit-Learn according to the data input.

We executed the offline phase of our proposal in which we evaluated the performance of the machine learning tools (the neural network and the polynomial regressor) as a function of the size of the dataset. The idea is to measure the
impact of the dataset size on the performance of the offline phase. For that purpose, we measure the RMSE for the regressor and the Accuracy, Type I, and Type II errors for the neural network (which works as a classifier).

Then, we evaluated our ADP algorithm in the online phase. We consider a horizon of one day and time stages of one hour each. We evaluated our proposal against 4 benchmarks aiming at comparing its performance with respect to the optimal solution (oracle) and assessing the gains associated with the use of ES and IC mechanism together. Finally, we measured the impact of the dataset size on the final performance of the ADP algorithm.

6.3 Results

In this section, we highlight the main results obtained in the paper. Table 6.1 shows the performance of the machine learning tools in the offline phase. We evaluate the impact of the size of the training set in the performance of the polynomial regressor and the neural network. This result shows that while the polynomial regressor estimates with low error the power consumption with a relatively small amount of samples (approximately 1000 samples), the neural network needs a more significant amount of samples.

Next, we evaluate the online phase. Figure 6.1 shows the aggregated power consumption in the network using our proposal, ADP ESIC, throughout one day, in time stages of one hour. This figure shows that our proposal achieves near-optimal power consumption and a big leap with respect to other benchmarks which do no combine ES and IC (ADP ES, Default configuration, Oracle ES). The combination of these two mechanisms allows our proposal to use fewer pico eNBs, reducing the power consumption. Figure 6.2 shows the ratio of UEs satisfying the QoS requirement during the evaluated day. Note that when the IC mechanism is deactivated (ADP ES, Default configuration, Oracle ES), the QoS threshold cannot be satisfied during the traffic peak (between stages 11 and 15) even when these benchmarks use high power consumption controls.

Table 6.2 shows the performance of the ADP algorithm as a function of the size of the training set used in the offline phase. When the training set is small, the estimations of the offline phase are inaccurate, which impacts negatively on the performance of the online phase. This table shows that while the energy consumption is almost not affected by the size of the training set, the QoS fulfillment requires as much samples as possible to attain an acceptable performance.

Finally, Fig. 6.3 illustrates the relation between the QoS threshold, $Q_{\text{min}}$, and the best achievable daily energy consumption (fulfilling this threshold). The higher the value of $Q_{\text{min}}$, the more the energy is necessary to satisfy the QoS requirement. This figure also shows that combining the ES and IC mechanisms attains the lowest energy consumption under any QoS requirement.
6. Article 4: Energy Saving and Interference Coordination in HetNets Using Dynamic Programming and CEC

<table>
<thead>
<tr>
<th># samples</th>
<th>RMSE (C)</th>
<th>Acc. (Q)</th>
<th>Typ. I err. (Q)</th>
<th>Typ. II err. (Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>263.80 ± 59.10</td>
<td>0.83 ± 0.03</td>
<td>0.13 ± 0.04</td>
<td>0.25 ± 0.09</td>
</tr>
<tr>
<td>300</td>
<td>13.08 ± 1.06</td>
<td>0.88 ± 0.01</td>
<td>0.09 ± 0.03</td>
<td>0.16 ± 0.05</td>
</tr>
<tr>
<td>1000</td>
<td>3.95 ± 0.14</td>
<td>0.92 ± 0.01</td>
<td>0.06 ± 0.02</td>
<td>0.13 ± 0.05</td>
</tr>
<tr>
<td>7000</td>
<td>3.11 ± 0.01</td>
<td>0.95 ± 0.01</td>
<td>0.04 ± 0.01</td>
<td>0.06 ± 0.02</td>
</tr>
<tr>
<td>18000</td>
<td>3.07 ± 0.02</td>
<td>0.95 ± 0.01</td>
<td>0.04 ± 0.01</td>
<td>0.05 ± 0.02</td>
</tr>
<tr>
<td>23000</td>
<td>3.08 ± 0.03</td>
<td>0.96 ± 0.01</td>
<td>0.03 ± 0.01</td>
<td>0.07 ± 0.02</td>
</tr>
</tbody>
</table>

Table 6.1: Performance of the machine learning tools as a function of the number of samples in the training set.

<table>
<thead>
<tr>
<th># samples</th>
<th>Daily energy consumption (kWh)</th>
<th>QoS fulfillment probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>29.95 ± 0.30</td>
<td>0.31 ± 0.11</td>
</tr>
<tr>
<td>300</td>
<td>28.70 ± 0.03</td>
<td>0.54 ± 0.12</td>
</tr>
<tr>
<td>1000</td>
<td>28.64 ± 0.00</td>
<td>0.59 ± 0.11</td>
</tr>
<tr>
<td>7000</td>
<td>28.64 ± 0.00</td>
<td>0.78 ± 0.05</td>
</tr>
<tr>
<td>18000</td>
<td>28.64 ± 0.00</td>
<td>0.85 ± 0.06</td>
</tr>
<tr>
<td>23000</td>
<td>28.64 ± 0.00</td>
<td>0.90 ± 0.05</td>
</tr>
</tbody>
</table>

Table 6.2: Performance of the ADP algorithm as a function of the number of samples in the training set.

Figure 6.1: Comparison of the aggregated instantaneous power consumption at each hour for $Q_{min} = 0.68$ under different policies.
Figure 6.2: Ratio of UEs satisfying the minimum required throughput at each hour for $Q_{\min} = 0.68$.

Figure 6.3: Accumulated energy consumption in one day for different values of $Q_{\min}$.

6.4 Contributions

The main contributions of the paper are summarized as follows:

- To consider the joint control of energy-saving and interference coordination mechanisms in HetNets leading to remarkable performance improvement
To formulate the problem as a constrained finite horizon MDP leveraging two facts: the traffic demands usually follow periodic patterns, and the knowledge and prediction of the user traffic load are crucial in order to select efficient network configurations.

To propose an ADP algorithm achieving near-optimal performance in terms of energy consumption and QoS of the users, reducing energy consumption up to 24% with respect to a fixed default configuration.

To build the ADP algorithm using a novel combination of stochastic control and machine learning techniques. In particular, we use Certainty Equivalent Control (CEC) in combination with Neural Networks and polynomial regression.

6.5 Conclusions

This paper presented a novel stochastic control approach addressing the energy-saving and the interference coordination problems jointly. It is applied to an LTE-A HetNet to minimize the energy consumption assuring QoS guarantees. Considering the periodic pattern of daily traffic demands, the problem is formulated as a finite horizon MDP where we apply the CEC principle for the sake of computational tractability. Specifically, we propose an ADP algorithm combining the CEC principle and machine learning techniques. We have shown that our proposal achieves near-optimal performance and a considerable improvement associated with addressing energy-saving and interference coordination problems jointly. In addition, our proposal allows a configurable QoS requirement.

There are, however, few aspects left for future research: First, we assume a daily periodic pattern in the traffic demands which can change depending on the day. For example, the traffic demands can change on weekends or during special events like sport games or concerts. In the case of regular weekends or vacation days, the prediction is straightforward. However, for special events, the traffic demands can change depending on the number of people, the traffic profile of the users, the type of event, etc. In order to address this issue, we can use external information (e.g., event program in the city) or detect the changes in the traffic patterns in an online fashion and adapt the algorithm to them.

Second, the selected time stage period is one hour. Although the traffic demands can be considered constant during this period, there are hours in which the traffic load changes rapidly. It would be interesting to investigate if a shorter time stage (higher frequency in selecting configurations) improves the performance. Using the proposal presented in this chapter, the offline phase has to be recomputed at any time we want to change the frequency in which the controls are selected.

Finally, our simulations have shown that the power activation cost of the pico eNBs does not have a high impact on the farsighted policy. This leads us to consider the use of myopic policies for this problem taking advantage of their lower computational complexity. This idea will be developed in the next chapters.
7.1 Summary

In the previous chapter, we presented the advantages of using Energy Saving (ES) and Interference Coordination (IC) mechanism together. We showed a great gain in terms of energy-saving with respect to the case in which these mechanisms are controlled separately. However, our simulations show that the impact of the power activation cost of the pico eNBs does not have a high impact on the performance, which leads us to propose a myopic policy in this chapter. Specifically, we formulate the problem as a contextual multi-armed bandit (also referred to as contextual bandit) \[82\].

The contextual bandit problem is a special case of the full reinforcement learning (RL) problem \[83\] with the following particularities: i) after each time stage we receive a reward associated with the current state and the selected control, i.e., the reward is not delayed; ii) the controls used in one time stage does not affect state transition, i.e., reward in the future is not affected by the selected controls in past time stages.

This formulation allows us to change the rate at which the controls are selected, that is, the duration of the time stage. We can even change the time stage duration depending on the velocity of the traffic load variation. In contrast, the ADP approach of previous chapter would needed to recompute the policy offline in case we wanted to change the rate at which the controls are selected.

In addition, this formulation does not assume any specific traffic pattern. This is an advantage especially in the case in which the traffic does not follow predefined patterns: concerts, sports games, etc.
Similarly to the previous chapter, we propose a semi-distributed scheme consisting of a global controller (learning agent) and local controllers, one for each macro eNB. The global controller manages a set of contiguous macro eNBs with a similar traffic profile selecting global controls which are sent to the local controllers. These global controls are translated by the local controllers into local control for each network sector.

The proposed learning agent at the global controller, referred to as ClassMAB \cite{84}, solves the contextual bandit problem and comprises two parts: a neural network (NN) classifier and a Multi-Armed Bandit (MAB) algorithm. The NN receives the traffic load in the network and a global control and decides if this pair is feasible in terms of QoS. This classifier allows us to obtain the set of QoS feasible controls for a given network state. Once we have the set of controls with QoS guarantees, we would like to find the control with the lowest power consumption belonging to this set. We propose the Control Constrained Descent (CCD) $\varepsilon$-greedy algorithm for this task. CCD $\varepsilon$-greedy is a novel and simple modification of the classic MAB algorithm referred to as Descent $\varepsilon$-greedy in which we take into account that not all controls are available to be selected at every time stage.

Both the NN classifier and the MAB algorithm are trained online while the network operates. That is, at the beginning of each time stage, the global controller selects one control to be used in the network. At the end of the stage, the global controller receives feedback in terms of power consumption and QoS, which allows the MAB and the NN to be trained.

Similarly to previous chapters, our proposal is evaluated in an LTE-A network simulator following the 3GPP guidelines. Our numerical results show that the proposed solution learns to progressively approach the best policy obtaining an energy-saving close to 20% with respect to a default policy and an improvement of 13% with respect to addressing ES and IC separately.

### 7.2 Methodology

We formulated the problem as a contextual multi-armed bandit intending to find a policy selecting efficient controls for each network context. As in the previous chapter, we divided the decision making into two levels (global and local controllers) in order to reduce the dimensionality of the state and control spaces.

We designed our proposal, ClassMAB, with two parts: a NN classifier and a MAB algorithm. It was implemented in Python from scratch. The NN was implemented using TensorFlow with 2 hidden layers comprising 50 units each.

We considered the same application scenario described in the previous chapter. The details of the network simulator can also be found in the previous chapter.

We considered 4 benchmark solutions to compare them against our proposal: i) the oracle policy, selecting the optimal global control at each time stage, ii) a default configuration considering the ES and IC mechanism deactivated, iii) Neural Bandit, a state-of-the-art online learning algorithm \cite{85}, and iv) ClassMAB (ES), the current proposal but only controlling the energy-saving mechanism.
We evaluated the performance of our proposal and the benchmark solutions in two phases: training and test phase. The training phase is composed of 1200 epochs of 200 time stages in which we evaluate the leaning velocity of the algorithms, and their performance after convergence. In the test phase, we evaluated the performance during one day after learning. In both training and test phases, the traffic was generated randomly according to a cyclic pattern. The selected performance metrics are the energy consumption, the ratio of QoS fulfillment, and the regret, which is the most commonly used metric for online algorithms.

7.3 Results

In this section, we highlight the main results of this work for both the train and test phases. Figure 7.1 shows the evolution of the cost over time in the training phase. Our proposal converges at epoch 350 and achieves the lowest operation cost with respect to the other solutions.

Figure 7.2 shows the accumulated regret during the training phase. The regret measures the losses incurred by a policy when it selects a suboptimal control, i.e., the closer the policy to the optimal one, the lower the increase of the regret. The slope of the regret achieved by our proposal tends to zero at the end of the training phase, which means that this policy is close to the optimal one.

Figure 7.3 shows the regret in the test phase. Note that, after training, our proposal incurs the lowest regret, far from the other benchmarks. In addition, it achieves an improvement of 19.48 % of energy savings with respect to the default configuration and a ratio of 0.982 of QoS fulfillment in the test phase. The benchmarks ClassMAB (ES) and NeuralBandit achieve an energy saving of 8.08 % and 13.26 %, respectively, and a ratio of QoS fulfillment below 0.98.

7.4 Contributions

The main contributions of this chapter are summarized as follows:

- An online learning framework for the joint control of energy-saving and interference coordination mechanisms in HetNets based on a contextual bandit problem.

- A novel approach to the above problem based on an online learning algorithm combined with machine learning tools. Specifically, our proposal relies on novel modification of a classic Multi-Armed Bandit algorithm and a neural network classifier.

7.5 Conclusions

This chapter presented a novel contextual bandit approach addressing energy-saving and interference coordination in LTE-A HetNets. In our proposal, the learning agent
Figure 7.1: Evolution of the cost function during the training phase.

Figure 7.2: Regret measured during the training phase.

(Global controller) learns efficient global configurations based on network performance samples and context information (global state of the network). These global configurations are interpreted locally by the Local Controllers on each network sector. Breaking down the decision into two levels allows the complexity of
The control space to grow linearly rather than exponentially, favoring the scalability of our mechanism. Our proposal is a novel combination of a neural network classifier and a MAB algorithm that is able to minimize the energy consumption providing QoS guarantees. Our numerical results show an energy-saving close to 20% with respect to a fixed default policy. The considerable improvement associated with addressing energy-saving and interference coordination problems jointly is also shown.

However, there are a few aspects of the work presented in this chapter that can be improved. First, the problem is formulated as a contextual bandit problem with the objective of minimizing a cost function. Depending on how this cost function is configured, the algorithm could violate the QoS requirement in exchange for a power consumption reduction. We consider that this is not desirable for the operator and, for that reason, it is more convenient to formulate the problem with restrictions.

Second, the proposed MAB algorithm does not take into account the context in the power consumption estimation. That is, there are variations in the power consumption of every control due to changes in the network traffic load. These changes in power consumption are considered noise for the MAB algorithm. Although this proposal has been shown to be effective, we believe that the performance can be improved if the context is taken into account in the learning of the power consumption.

Finally, the operator could desire a quicker convergence in order to avoid the network to be operating under configurations with low performance. The learning time (in terms of the number of samples) of a neural network cannot be character-
ized in general and depends on its architecture. Although our proposal presents a reasonable convergence time, we believe that leveraging the threshold structure of the QoS restriction can improve the convergence time. The next chapter develops these ideas.
8.1 Summary

In this chapter, we improve the proposal of the previous one in several aspects detailed as follows. We formulate the problem as a constrained contextual multi-armed bandit. This is a novel variation of the contextual bandit problem in which the optimal policy always has to satisfy the problem constraints. Following this formulation, we propose a new solution referred to as Bayesian Response Estimation and Threshold Search (BRETS) [86], which comprises two parts.

The first one, called Threshold Search (TS) is a new mechanism based on the premise that, for each control action, the QoS objective is fulfilled only when the UE traffic intensity is below some threshold. The traffic threshold for each control is initially unknown, and the efficient discovery of all the thresholds is the objective of TS.

The second learning process, Response Estimation, aims at estimating the function that maps traffic intensity to network energy consumption (the network response) for each control. Response Estimation uses a Bayesian approach, in particular, a Gaussian Process, which generates response function estimations with relatively few samples and provides the uncertainty of the estimation at each traffic intensity. This allows us to apply the principle of optimism in the face of uncertainty used by classic bandit algorithms [74, 87] in a novel way: each control action is associated with a function instead of a scalar.

The combination of these two parts results in a computationally efficient solution with a much higher learning rate with respect to other alternatives. In addition, BRETS comes with a bound on its expected convergence time. Our numerical results show that our proposal is only 1% less efficient than an ideal oracle policy, clearly outperforming other benchmark algorithms.
8.2 Methodology

The methodology in this chapter is quite similar to the one described in the previous chapter. The main differences are highlighted as follows. The problem was formulated as a constrained contextual multi-armed bandit in which the QoS requirement is considered a hard constraint.

The proposal in this chapter, referred to as BRETS, was designed in two parts: the Bayesian response estimation which uses Gaussian Processes, and the threshold search which is a novel contribution of this work. We supported our proposal with theoretical results providing an upper bound on the expected convergence time.

The proposal was implemented using Python. For the Gaussian Processes, we used the Python library Scikit-Learn. The mean of its prior distribution is set to zero, and the covariance matrix is specified by the Matern kernel. The hyperparameters of the kernel are optimized using a quasi-Newton algorithm. For the sake of computational efficiency, these hyperparameters are updated only 10 times during the training phase, which suffices to get the highest performance.

The application scenario and the network simulator were the same as the ones in the two previous chapters. In the set of benchmark solutions, we included the proposal described in the previous chapter, ClassMAB. Finally, in order to provide statistical significance, we obtained all the results in this work averaging 30 independent simulation runs.

8.3 Results

As we explained in the previous chapter, we perform the evaluation in two phases: training and test phase. Figure 8.1 shows the evolution of the cost during the training phase. We can observe that our proposal, BRETS, only needs a few epoch to converge, much faster than any other benchmark. In addition, it operates at a smaller cost, as shown in the augmented section in Figure 8.1.

In the test phase, we evaluate a one day period using a time stage duration of 10 minutes, that is, we evaluate a total of 144 stages. Figures 8.2 shows the power consumption during the test phase. The power consumption achieved by BRETS is really close to the optimal one, while the benchmarks present a considerable gap. Figures 8.3 shows the probability of failing the QoS during the test phase. BRETS achieves probability zero at every time stage. The benchmarks not considering IC (Default configuration and ClassMAB (ES)) are not able to meet the QoS requirement in the traffic peak, that is, they fail the QoS requirement with probability one in some stages.

Finally, table 8.1 summarizes the results of our proposal and all the benchmark solutions in the test phase. Note that our proposal is only 1% less efficient than the oracle policy in terms of energy consumption and always meeting the QoS requirement.
8.4 Contributions

The main contributions of this work are:

- An online learning framework for the joint control of energy-saving and interference coordination mechanisms in HetNets based on a new variant of contextual bandit problems incorporating constraints on an additional performance metric.
Figure 8.3: Probability of failing the QoS requirement at each stage of test phase.

<table>
<thead>
<tr>
<th>Method</th>
<th>Energy saving (%)</th>
<th>Ratio of QoS fulfillment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>25.701%</td>
<td>1</td>
</tr>
<tr>
<td>Default conf.</td>
<td>0%</td>
<td>0.9652</td>
</tr>
<tr>
<td>NeuralBandit</td>
<td>13.258%</td>
<td>0.9777</td>
</tr>
<tr>
<td>ClassMAB (ES)</td>
<td>8.085%</td>
<td>0.9652</td>
</tr>
<tr>
<td>ClassMAB</td>
<td>19.483%</td>
<td>0.9817</td>
</tr>
<tr>
<td>BRETS</td>
<td>25.007%</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 8.1: Summary of numerical results in the test phase

- A novel approach to the above problem based on associating two learning processes to each control, one per performance metric. Each process aims at learning a function mapping the context variable (network traffic intensity) to one performance metric.

- A new algorithm (Threshold Search) implementing the learning process associated with the QoS response of the system. This algorithm is characterized in terms of the expected convergence time.

8.5 Conclusions

This paper presented an online learning algorithm for jointly controlling the energy-saving and interference coordination mechanisms in HetNets. Our framework considers two levels. At a higher level, a global controller selects global controls applicable to a cluster of contiguous macro eNBs. At the lower level, local con-
controllers decide how to apply the prescribed global control on each macro eNB sector. The proposed learning algorithm, running in the global controller, addresses a constrained contextual bandit problem using a novel approach: for each action, the algorithm handles two simultaneous learning processes, one for the performance objective (network energy consumption) and another for the constraint (QoS fulfillment). Each one of these processes aims at learning a function, instead of a scalar as in conventional bandit algorithms. These functions map the context (network traffic intensity) to the performance metric of interest. For the QoS function, we propose a novel strategy for finding the traffic threshold below which each control fulfills the QoS objective. For estimating the energy consumption functions, the algorithm uses Gaussian processes, which allows the BRETS to balance exploration and exploitation when selecting controls. In our numerical simulations, BRETS outperformed other alternatives attaining an energy saving of 25% and a QoS fulfillment ratio of 100%, very close to the results obtained by an oracle policy.

Future work includes the following lines. Our proposal could be complemented by an auxiliary learning algorithm for finding clusters of adjacent cells in which joint IC-ES controls can be applied effectively, i.e., for assigning cells to global controllers. The fewer the sectors per cluster, the more similar the traffic profile among clusters, which can increase the performance of our proposal, at the cost of increasing the mutual interference among clusters and requiring more instances of the ADP controller. On the other hand, larger clusters lead to more variance in the traffic intensity of the sectors but reducing the mutual interference among clusters. Another alternative to reduce mutual interference among cluster boundaries is to coordinate the learning periods between them. A possible approach would be to alternate the learning periods of adjacent clusters, at the cost of increasing the convergence time.
Part II

Article list
Online Learning for Interference Coordination in Heterogeneous Networks

Jose A. Ayala-Romero, Juan J. Alcaraz, Javier Vales-Alonso, Esteban Egea-López
Universidad Politécnica de Cartagena
TIC Department
Plaza del Hospital, 1, Cartagena, Spain
Email: {josca.ayala, juan.alcaraz, javier.vales, esteban.egea}@upct.es

Abstract—This paper focuses on interference coordination between the small cell and macro cell tiers of a wireless access network. We propose a new perspective based on a model-free learning strategy, not requiring any previous knowledge about the network (e.g., topology, interference graph, scheduling algorithms). Our approach is based on a stochastic optimization algorithm known as Response Surface Methodology, that we use to learn the optimal parameter configuration during network operation (online learning) adapting to changes on network conditions (e.g., traffic, user positions). The result is a simple, effective and flexible mechanism that outperforms previous proposals. As a case study we apply our scheme to the dynamic adjustment of LTE-A eICIC parameters (CRE bias and ABS ratio).

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

I. INTRODUCTION

Small cells are considered one of the key technologies in current and future 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, and has been a hot research topic for the last few years. In this paper we propose a new strategy to address this problem. In contrast to previous works, our proposal is based on a model-free online learning formulation. To illustrate our approach, we describe its application to the advanced Inter Cell Interference Coordination (eICIC) functionality of Long Term Evolution Advanced (LTE-A) networks.

The interference between macro cells (macro eNBs) and small cells (pico eNBs) is determined 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 [2]–[5]. Figure 1 shows the 5th percentile throughput (a usual metric for LTE-A) [6]

![Image of a graph showing throughput] (7) obtained by simulating a scenario defined according to 3GPP guidelines [8]. Both surfaces were obtained for the same network layout, but using two different traffic demands. The optimal configurations of (ABS ratio, CRE bias) for 10 Mbps and 75 Mbps per macro sector are (3/8, 9 dB), (4/8, 12 dB), respectively. Therefore, as traffic conditions change, the eICIC parameters should be dynamically reconfigured.

Motivated by this requirement, this paper proposes a mechanism based on a stochastic optimization algorithm called Response Surface Methodology (RSM) [9]–[11], which learns the optimum configuration online, adapting to the network changes. The rest of the paper is organized as follows. The related works and contributions are discussed in Section II. The problem is formulated in Section III, and Section IV presents our RSM-based mechanism. In Section V we describe the simulation model used to evaluate our proposal, and numerical results and benchmark comparisons are provided in Section VI.

II. RELATED WORK AND CONTRIBUTION

A. Related Work

There is a broad literature on interference management for HetNets. However, we adopt a new perspective. We consider the variable nature of the network as a basic element of the problem formulation, in contrast to [2]–[5] that consider a static network situation, and do not address explicitly the real-time adaptation to dynamic network conditions (traffic intensity, number of users, user positions).

While some works address the problem considering dynamic network conditions as well [6], [12], their approach
differ to ours in some crucial aspects. First, they propose heuristic mechanisms that are very scenario-dependent. For example, although [6] performs well in the usual LTE-A scenarios for which it is designed, it is not adequate for highly dense scenarios. Second, these schemes only focus on optimizing the ABS ratio while, as shown in the introduction, both eICIC parameters have a joint impact on performance. Our mechanism can optimize two or more parameters simultaneously, independently on the scenario.

Therefore, we essentially follow a model-free optimization approach. This also contrasts with many previous works relying on mathematical models of the system [2] [3] [5]. For mathematical tractability, even the most complex models imply simplifications and assumptions that limit their application in real operating networks. For example, stochastic geometry [13] [14] is an excellent tool for high-level evaluation like determining the capacity bounds of a network. But it assumes that nodes are deployed according to spatial distributions like Poisson point process, while a fine-tuned parameter configuration should be done for the real network layout.

Finally, our approach requires low overhead in terms of computation and signaling. For example, some mechanisms require retrieving information about the network topology [2], and the interference graph of the network [3]. Our proposal only collects performance samples. Table I summarizes the characteristics of others works in comparison to ours.

### B. Contribution

This paper proposes a novel self-organizing strategy for HetNets, capable of learning the optimal configuration of interference coordination parameters. Our mechanism is built upon a stochastic optimization algorithm known as Response Surface Methodology. The main novelty is the adoption of a model-free learning approach in interference management, not requiring any previous knowledge about the network topology, interference graph, radio propagation conditions, and so forth. As a result we obtain a simple, effective and adaptive algorithm that outperforms previous proposals. As a case study we apply our proposal to the dynamic adjustment of LTE-A eICIC parameters (CRE bias and ABS ratio).

### 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) 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 avoids that the lower transmission power of the pico eNBs cause the underuse of their resources. 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).

**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. We consider a cluster of macro eNBs sharing the same ABS pattern (synchronized muting) and the same CRE bias [15], [7], [14]. The cluster is assumed to cover a geographical area with a homogeneous traffic profile.

#### B. Stochastic Optimization Problem Formulation

Let us consider an operator who wants to optimize certain performance metric, referred to as \(F\), depending on the interference coordination parameters of the system. In eICIC these parameters are the ABS ratio \((\gamma)\) and CRE bias \((\phi)\). Because of the random 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 a configuration \(\gamma\) and \(\phi\) that maximizes \(E[F(\gamma, \phi)]\).

Let \(\mathcal{X} = (\gamma, \phi)^T\) denote the eICIC configuration. For the ABS ratio we allow the system to randomize between contiguous ABS patterns, so that \(\gamma\) can take continuous values. The convex set \(\mathcal{P} = \{x = (\gamma, \phi)^T: 0 \leq \gamma \leq 1, 0 \leq \phi \leq \phi_{\text{max}}\}\) contains all the feasible values of \(x\). We can formulate the main problem as

\[
\max_{x \in \mathcal{P}} E[F(x)].
\]
We are assuming that the operator has full freedom to select the performance metric $F$. It can be associated to the physical layer (e.g. signal to interference and noise ratio, SINR) or to upper layers (e.g. data throughput). Given the amount and the complexity of the stochastic processes involved, a mathematical expression to estimate $E[F(x)]$ accurately for a real operating network is, in general, intractable.

C. Online Learning Approach

Because the objective function $E[F(x)]$ is unknown, our proposal relies on learning the optimal configuration by obtaining samples of $F(x)$ from the operating network itself. This data-driven approach is known as online learning. Let us reformulate (1) within this framework.

An online learning algorithm operates in countable time stages $k = 0, 1, \ldots$. At each stage $k$, the algorithm selects a configuration $x_k$ and obtains an observation of the performance $y_k = F(x_k, \omega_k)$ where $\omega_k$ represents the (random) state of the system at stage $k$, comprising all the random variables affecting $F$ (e.g. traffic demands, UEs positions, propagation conditions). The configuration $x_k$ has to be selected according to the history of the system up to $k - 1$, defined as the set of past configurations and observations: $H_{k-1} = \{x_0, y_0, \ldots, x_{k-1}, y_{k-1}\}$. The objective, up to a given stage $N$, is to solve

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

(2)

Because no previous knowledge about the system is assumed, and $\omega_k$ is not included in the observations, the algorithm addressing the above problem follows a black box optimization approach. This type of algorithm needs to explore the objective function $F$, that is, to select configurations $x_k$ that might not be considered as the best option given the history of the system, 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 $\omega_k$ 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 would not be optimal in future stages because the statistical conditions for which $x^*$ is optimal are no longer present. Therefore, the algorithm should be able to adapt the system configuration to long term changes in user activity.

IV. STOCHASTIC OPTIMIZATION ALGORITHM

A. Response Surface Methodology

Let us consider an algorithm to find the optimal solution of the original problem (1) 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 should converge 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

$$x^{(n+1)} = x^{(n)} + \alpha d(x^{(n)})$$

(3)

where $d(x^{(n)})$ is the steepest-ascent search direction at point $x^{(n)}$, and $\alpha$ is the step-size weighting factor.

Let us define $f(x) = E[F(x)]$. When $f$ is a differentiable function, the steepest-ascent direction at point $x^{(n)}$ is given by the gradient of $f$, i.e. $d(x^{(n)}) = \nabla f(x^{(n)})$. However, since $f$ is unknown we cannot assess if it is differentiable nor compute $\nabla f(x^{(n)})$. We can overcome this issue by generating an approximate steepest ascent direction $\hat{d}(x^{(n)})$ such that the iteration (3) provides, with high probability, better configurations as $n$ increases. In order to compute $\hat{d}(x^{(n)})$, the algorithm must take samples of the performance metric during network operation. Response Surface Methodology (RSM) [9]–[11] provides an efficient way to sampling the system and computing $\hat{d}(x^{(n)})$. In this section we detail how we can use RSM as an online learning algorithm.

Let $k$ denote the number of stages that the system has been operating when the gradient-ascent algorithm obtains the $n$-th iterate $x^{(n)}$ (note that, in general, it takes several stages to perform one gradient-ascent iteration). Therefore, $x^{(n)}$ is the outcome of the learning algorithm given the current system history $H_k = \{x_0, y_0, \ldots, x_k, y_k\}$.

In order to obtain next iterate $x^{(n+1)}$ using (3), RSM generates 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 $\theta$. For $x \in \mathbb{R}^2$, the parameter vector is $\theta = (\theta_1, \theta_2)^T$, and the approximation of $f$ is given by

$$\hat{f}(x|\theta) = \theta_0 + x^T \theta_1 + \beta$$

(4)

where $\beta = (\beta_1, \beta_2)^T$.

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

1) Select a set $D^{(n)}$ of design points in the vicinity of $x^{(n)}$.
2) Take a (possibly random) number $p$ of samples using the points in $D^{(n)}$. The set of samples $S^{(n)}$ comprises points taken from $D^{(n)}$ and their corresponding observations: $S^{(n)} = \{x_{k+1}, y_{k+1}, \ldots, x_{k+p}, y_{k+p}\}$. The set $S^{(n)}$ is therefore the incremental history between two consecutive gradient-ascent iterations, $S^{(n)} = H_{k+p} - H_k$.
3) Using $S^{(n)}$, construct matrix $W^{(n)}$ and vector $y^{(n)}$ as follows

$$W^{(n)} = \begin{bmatrix} 1 & x_{k+1}^T \\ 1 & x_{k+2}^T \\ \vdots & \vdots \\ 1 & x_{k+p}^T \end{bmatrix}, \quad y^{(n)} = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+p} \end{bmatrix}.$$  

(5)

4) Compute the least squares estimation of $\hat{\theta}^{(n)}$:

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

(6)
Once $\theta^{(n)}$ is obtained, iteration (3) is applied with the following estimation of the steepest-ascent direction

$$\hat{d}(x^{(n)}) = \nabla \hat{f}(x^{(n)}|\theta^{(n)}) = \theta^{(n)}_{x=0}. \tag{7}$$

### B. Sampling Strategy

Let us see how to determine, at each iteration, the set $D^{(n)}$ and the number of samples $p$ with statistical guarantees. Since we aim at approximating $f$ at current iterate $x^{(n)}$, the design points should be small perturbations of $x^{(n)}$. 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 [10], 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

$$D^{(n)} = \{ x^{(n)} \}
\begin{align*}
&x^{(n)} + (\delta_1, \delta_2)^T, \\
&x^{(n)} + (-\delta_1, \delta_2)^T, \\
&x^{(n)} + (\delta_1, -\delta_2)^T, \\
&x^{(n)} + (-\delta_1, -\delta_2)^T.
\end{align*} \tag{8}$$

Thus, $\delta = (\delta_1, \delta_2)$ 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 biased gradient estimations. In our case study, $\delta$ configuration is based on experience (previous works and simulations) [9] [10]. An automatic adjustment of $\delta$ and $\alpha$ is left for further study.

If we take the same number of samples from each point in $D^{(n)}$, the columns of $W^{(n)}$ will be orthogonal and will have the same number of positive and negative perturbations. This property guarantees that $(W^{(n)})^T W^{(n)}$ in (6) is invertible.

The search direction is determined by the regression coefficients $\hat{\theta}^{(n)}$ obtained from noisy samples of $f$. We can characterize statistically these coefficients to evaluate the accuracy of the search direction. Specifically, given $\hat{\theta}^{(n)}$, we can obtain the angle $\varphi$ within which the real steepest-ascent direction is contained with a given confidence degree $C$. The algorithm can take samples until $\varphi$ is smaller than a desired target angle $\varphi^*$. We propose to use $\varphi^*$ and $C$ as design parameters that dynamically determine the amount of samples $p$ taken during each iteration $n$.

Let $d = |D^{(n)}| (= 5$ in our case study). After obtaining $x^{(n)}$, the algorithm takes $p = d$ samples, one at each point in $D^{(n)}$, builds $W^{(n)}$ and $y^{(n)}$ with these samples, and then uses (6) to obtain $\hat{\theta}^{(n)}$. The variance of $f(x|\hat{\theta}^{(n)})$ is estimated by

$$\hat{\sigma}^{(n)} = \frac{1}{p-2} \| y^{(n)} - W^{(n)} \hat{\theta}^{(n)} \|^2 \tag{9}$$

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

$$\hat{\Sigma}^{(n)} = \left( W^{(n)} \right)^T W^{(n)} \hat{\sigma}^{(n)} \tag{10}.$$

Here, the $j$-th element on the diagonal of $\hat{\Sigma}^{(n)}$ is the variance of the $j$-th element of $\hat{\theta}^{(n)}$ [16]. Let us denote by $s_{\theta}^2$ the maximum variance of $\theta_{x=0}$.

In order to obtain the angle $\varphi$ for a confidence degree $C$, we can use the following expression from [9]

$$\frac{\varphi}{2} = \arcsin \left( \frac{s_\theta^2 F_C(1, p - 2)}{\sum_{j=1}^{p} (\hat{\theta}^{(n)}_j)^2} \right)^{1/2} \tag{11}$$

where $F_C(1, p - 2)$ is the $C$-th percentile of the $F$-distribution.

Therefore, if $\varphi > \varphi^*$, then $d$ new samples are taken from each point in $D^{(n)}$ and incorporated to $W^{(n)}$ and $y^{(n)}$, $p$ is updated by $p \leftarrow p + d$, and $\theta^{(n)}$ is recomputed with (6). The resulting RSM algorithm with confidence in the search direction is shown in Algorithm 1, and illustrated in Figure 2. Note that, when $x^{(n)}$ is close to the optimum, $\| \theta^{(n)} \|$ approaches 0 and subsequent iterations remain within the region of the optimum (exploitation) while still monitoring the gradient. When the optimum drifts to a different location, then $\| \theta^{(n)} \|$ increases, making the algorithm re-start learning.

### Algorithm 1: RSM with confidence in the search direction

1: Select initial point $x^{(0)}$
2: for $n = 0, 1, 2, \ldots$ do
3: Generate $D^{(n)}$, set $p = 0$, and set $\varphi$ to a value greater than $\varphi^*$
4: while $\varphi > \varphi^*$ do
5: Obtain one sample at each point of $D^{(n)}$ ($d$ new samples)
6: Insert the samples in $W^{(n)}$ and $y^{(n)}$
7: $p \leftarrow p + d$
8: Estimate $\hat{\theta}^{(n)}$ using (6)
9: Estimate $\varphi$ using (9), (10), and (11)
10: end while
11: $x^{(n+1)} = x^{(n)} + \alpha \hat{\theta}^{(n)}$
12: end for

### V. DESCRIPTION OF THE SIMULATED SCENARIO

This section describes the LTE-A scenario where we have evaluated our mechanism. It is based on the 3GPP guidelines [8]. 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. All the macro eNB sectors contain the same number of pico eNBs. These pico eNBs are uniformly distributed within each sector. We simulate the 7 central macro eNBs (21 sectors) while the external ones act as an inter-cell interference source, replicating the effect of a larger network. The inter-site distance is 500 m and the total bandwidth of the system is 10 MHz. Table II summarizes the remaining physical parameters.

The traffic model is defined by the following elements:
- At the user level we apply the FTP traffic model [8]. According to this model each incoming user has to download one file and stays in the system until the file is fully downloaded. The file size is 0.5 Mbytes.
- Users arrive according to a Poisson process. The parameter $\lambda$ provides the average number of arrivals per second (UE/s or equivalently file/s).
- Each incoming user is dropped either at a uniformly distributed position over the macro eNB area, or in a cluster around a pico eNB. The first case occurs with probability $P_{\text{macro}}$, and the second one with $P_{\text{pico}} = 1 - P_{\text{macro}}$.

Although Proportional Fair (PF) is used as the scheduling policy in some related works [6] [4] [7], other works like [17], have shown that PF is not the best choice when eICIC is used, because ABS subframes should be preferably assigned to users in the CRE regions, while PF cannot discriminate among UEs. For this reason we adopt the modification proposed in [17] for eICIC.

VI. NUMERICAL RESULTS
A. Learning Rate and Adaptability
Let us consider a plug-and-play situation where the learning algorithm starts to operate with an initial eICIC configuration $x^{(0)}$, possibly far from the optimal one. The algorithm parameters are set to $\delta = (0.035, 0.5)$, $\alpha = 0.035$, $\varphi^* = 8^\circ$, $C = 0.95$. The performance metric $y_k$ is the 5th percentile throughput. Figure 3 shows the eICIC parameters ($\gamma$ and $\phi$) generated by the algorithm over consecutive steepest-ascent iterations, and the evolution of the average performance obtained with a sliding window of 60 samples. Around the average performance we depict the region containing 90% of the performance samples. The width of this region reflects the variance of the samples and the effects of the exploration of the algorithm.

We see that the algorithm approaches the optimum performance in roughly 25 steepest-ascent iterations. The number of samples per iteration depends directly on the setting of $\varphi^*$ and $C$. For the selected values ($\varphi^* = 8^\circ$, $C = 0.95$), each steepest-ascent iteration needs 17.5 samples on average. For $\varphi^* = 15^\circ$ this number reduces to 6.6, although less precision on the angle estimation could increase the number of required iterations.

To illustrate how the algorithm adapts to network variations, Figure 4 shows a change on the traffic conditions from $\lambda = 18.75$ UE/s, $P_{\text{pico}} = 9/10$ to $\lambda = 7.5$ UE/s, $P_{\text{pico}} = 1/10$, the corresponding adaptation of $x^{(n)}$, and the evolution of the performance measurements.

B. Comparison to benchmark solutions
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 provides a baseline reference for eICIC self-configuration schemes.
- Optimal configuration for fixed traffic conditions. We consider the case where the operator configures its eICIC parameters assuming static traffic conditions, following for example a worst-case approach or assuming the most usual situation.
- Proportional-Fair-Based ABS (PF-ABS). This state-of-the-art heuristic proposed in [6] modifies $\gamma$ according to the average of the Proportional Fair indexes at each eNB. This average, updated on a frame-by-frame basis, is used as an indicator of whether the eNB performance is improving or deteriorating, allowing PF-ABS to equalize the performance by redistributing the resources (subframes) among the macro and pico eNBs. In contrast to our mechanism, PF-ABS does not include CRE bias control.
<table>
<thead>
<tr>
<th>Frame structure</th>
<th>Subframe 1 ms, Protected-subframe pattern 8 ms, Frame 10 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit power</td>
<td>Macro eNB: 46 dBm, pico eNB: 30 dBm</td>
</tr>
<tr>
<td>Macro eNB antenna</td>
<td>Pattern: ( A_T \propto -\min \left{ 12 \left( \frac{30}{30 + d} \right)^2, 26dB \right} ) Gain: 14 dB</td>
</tr>
<tr>
<td>Pico eNB antenna</td>
<td>Pattern: Omnidirectional, Gain: 5 dB</td>
</tr>
<tr>
<td>Macro to UE path loss</td>
<td>( 128.1 + 37.6 \cdot \log_{10}(R/\text{Km}) ) where ( R ) is the macro eNB to UE distance</td>
</tr>
<tr>
<td>Pico to UE path loss</td>
<td>( 149.7 + 36.7 \cdot \log_{10}(R/\text{Km}) ) where ( R ) is the pico cell to UE distance</td>
</tr>
<tr>
<td>Shadow fading</td>
<td>Lognormal distribution with 10 dB standard deviation</td>
</tr>
</tbody>
</table>

**TABLE II**

**SIMULATION PARAMETERS**

**VIII. ACKNOWLEDGEMENTS**

This work was supported by project grant MINECO/FEDER AIM TEC2016-76465-C2-1-R. Jose Antonio Ayala-Romero also acknowledges personal grant FPU14/03701.

**REFERENCES**


For the sake of comparison we assume that PF-ABS operates with optimal CRE.

Figure 5 shows the average performance for traffic intensities ranging from 10 to 130 Mbps per macro sector. One of the reasons why our scheme outperforms PF-ABS is that in PF-ABS each macro eNB has its own ABS pattern (unsynchronized muting) as a result of their local PF indexes, while our mechanism maintains a global ABS pattern (synchronized muting). Under unsynchronized muting, the ABS subframes present higher interference from other macro eNBs which may deteriorate the performance of the UEs located at CRE regions. For higher \( P_{\text{ABS}} \) or more pico eNBs per sector (dense scenarios), this effect becomes more relevant because more users fall in the CRE regions.

**VII. CONCLUSIONS**

We have presented a novel perspective to address interference coordination between macro and small cells in HetNets. Our premises are that a real operating network is too complex to predict its performance accurately, and its response to control parameters changes over time. Therefore, we propose a model-free scheme capable of learning the optimal configuration online. Because our approach is data-driven instead of model-based, our mechanism can optimize any performance metric independently of the network layout and state. We have evaluated our scheme for the self-configuration of LTE-A eICIC parameters, obtaining promising results.
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 spatial 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.

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 5th 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...
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 blackbox 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 5th 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
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, ..., 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 ($\gamma$) and CRE bias ($\phi$). 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

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

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

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

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\textsuperscript{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\textsuperscript{th} percentile throughput is defined as the value below which 5\% of the throughput samples fall.\textsuperscript{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 \( \omega_k \) 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

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

Because no previous knowledge about the system is assumed, and \( \omega_k \) 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 \( \omega_k \) 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.

### 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 \( x^{(n)} \), \( \alpha \) 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 \( d(x^{(n)}) \).

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 \( d(x^{(n)}) \) 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 \( \theta \). 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

\[
\hat{f}(x|\theta) = \theta_0 + x^T \theta - \theta_0 \tag{5}
\]

where \( \theta_0 = (\theta_0, \theta_1)^T \).

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 \( S^{(n)} \): contains a (possibly random) number \( p \) of configurations taken from \( \mathcal{D}^{(n)} \) during iteration \( n \), and their performance observations: \( S^{(n)} = \{ x_k, y_k, \ldots, x_{k+p-1}, y_{k+p-1} \} \). The set \( S^{(n)} \) is therefore the incremental history between two consecutive steepest-ascent iterations, \( S^{(n)} = \mathcal{H}_{n+p} - \mathcal{H}_n \). Note that \( p > D \), because it might be necessary to take more than \( D \) samples to generate a statistically reliable approximation of \( f \).[1]

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

1In our simulated scenario the system obtains one sample of the 5\textsuperscript{th} percentile throughputs from 1000 throughput samples
1) Generate the set $D(n)$ of design points in the vicinity of $x^{(n)}$.
2) Make observations of the performance $F$ at the configurations in $D(n)$, and store the samples in $s_n$.
3) Construct matrix $W(n)$ and vector $y(n)$ from the configurations and observations in $s_n$ respectively, as follows
   \[ W(n) = \begin{pmatrix} 1 & x_{*1}^T \\ 1 & x_{*2}^T \\ \vdots & \vdots \\ 1 & x_{*p-1}^T \end{pmatrix}, \quad y(n) = \begin{pmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+p-1} \end{pmatrix}. \] (6)
4) Compute $\theta(n)$ using least squares:
   \[ \theta(n) = \left( \left( W(n)^T W(n) \right)^{-1} \right) W(n)^T y(n). \] (7)

Once $\theta(n)$ is obtained, iteration (4) is applied with the following estimation of the steepest-ascent direction
   \[ \hat{d}(x^{(n)}) = \nabla \hat{f}(x^{(n)}) = \theta(n). \] (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 $x_k$, $x_{k+1}$, ..., and gathers the observations taken form the network.

B. Sampling Strategy

Let us see how to determine, at each iteration, the set $D(n)$ 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
   \[ g^{(n)} = \{ x^{(n)} + (\delta_1, \delta_2)^T, x^{(n)} + (-\delta_1, \delta_2)^T, x^{(n)} + (\delta_1, -\delta_2)^T, x^{(n)} + (-\delta_1, -\delta_2)^T \}. \] (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 $g^{(n)}$, the columns of $W(n)$ will be orthogonal and will have the same number of positive and negative perturbations. This property guarantees that $\left( W(n)^T \right)^2 W(n)$ 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 $\varphi$ 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 $\varphi$ is smaller than a desired target angle $\varphi^*$. After obtaining $x^{(n)}$, the algorithm takes $D$ samples (i.e. initially $p = D$), one at each point in $g^{(n)}$, builds $W(n)$ and $y(n)$ with these samples, and then uses (7) to obtain $\theta(n)$. The variance of $f(\theta^{(n)})$ is estimated by
   \[ \left( \hat{\sigma}^2 \right)^2 = \frac{1}{p-2} \| y(n) - W(n) \theta(n) \|^2 \] (10)
and the covariance matrix of $\theta(n)$ is given by
   \[ \hat{\Sigma}^2 = \left( W(n)^T W(n) \right)^{-1} \left( \hat{\sigma}^2 \right)^2. \] (11)

Here, the $j$-th element on the diagonal of $\hat{\Sigma}$ is the variance of the $j$-th element of $\theta^{(n)}$ [30]. Let us denote by $\hat{\sigma}^2$ the maximum variance of $\theta$. In order to obtain the angle $\varphi$ for a confidence degree $q$, we can use the following expression [10]
   \[ \frac{\varphi}{2} = \arcsin \left( \frac{\hat{\sigma}^2 \mathcal{T}_q(1, p - 2)}{\sum_{j=1}^{\infty} \left( \theta^{(n)} \right)^2} \right)^{1/2} \] (12)
where $\mathcal{T}_q(1, p - 2)$ is the $q$-th percentile of the F-distribution.

Therefore, if $\varphi > \varphi^*$, then $D$ new samples are taken from each point in $g^{(n)}$ and incorporated to $W(n)$ and $y(n)$. $p$ is updated by $p = p + D$, 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**

1: Select initial point $x^{(0)} \in \mathcal{P}$
2: for $n = 0, 1, 2, \ldots$ do
3: Generate $g^{(n)}$, set $p = 0$, and set $\varphi$ to a value greater than $\varphi^*$
4: while $\varphi > \varphi^*$ do
5: Obtain one sample at each configuration in $g^{(n)}$ ($D$ new samples)
6: Insert the samples in $W(n)$ and $y(n)$
7: $p = p + D$
8: Estimate $\theta^{(n)}$ using (7)
9: Estimate $\varphi$ using (10), (11), and (12)
10: end while
11: $x^{(n+1)} = \Pi_{\mathcal{P}} \left( x^{(n)} + \alpha^{(n)} \hat{d}(x^{(n)}) \right)$
12: 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 prove the convergence of the stochastic iteration $x^{(n+1)} = x^{(n)} + \alpha^{(n)} \hat{d}(x^{(n)})$, using stepsizes that are square-summable ($\sum_{m=0}^{\infty} \alpha^{(n)} = \infty$) but not summable ($\sum_{m=0}^{\infty} \alpha^{(n)} < \infty$). One key requirement of this result is $\hat{d}(x^{(n)})$ being a sufficiently accurate estimator of $d(x^{(n)})$.
which is precisely what Algorithm 1 pursues with the statistical guarantees imposed on $\varphi$.

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 $\alpha$. 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 eCIC 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 eCIC parameter configuration problem comprises now $r$ stochastic constraints

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

where $l = (0, 0)^T$ and $u = (1, \varphi_{\text{max}})^T$, 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** $d(x^{(n)})$ 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-exploitation, 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 $\theta_i$. 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 $\hat{f}_i(x|\theta_i) = \theta_{i,0} + x^T \theta_{i,1}$.

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: $x_{k,i} = F_i(x_k, w_k)$, $i = 0, \ldots, r$, which are stored in their corresponding sets of samples $S_k^{(n)}$. 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

$$
\hat{\beta}_i^{(n)} = \left( W_i^{(n)}^T W_i^{(n)} \right)^{-1} \left( W_i^{(n)} \right)^T y_i^{(n)}, \quad i = 0, \ldots, r - 1.
$$

(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 $c^{(n)}$:

$$
\Theta^{(n)} = \begin{pmatrix}
\hat{\beta}_1^{(n)} & \hat{\beta}_2^{(n)} \\
\vdots & \vdots \\
\hat{\beta}_r^{(n)} & \hat{\beta}_r^{(n)} \\
\end{pmatrix}, \quad c^{(n)} = \begin{pmatrix}
\alpha_1 - \hat{\beta}_1^{(0)} \\
\vdots \\
\alpha_r - \hat{\beta}_r^{(0)}
\end{pmatrix}.
$$

(15)

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

$$
\max_{x,v,z} \Theta_{0,0}^{(n)} x^T x \\
\text{s.t.} \quad \Theta_{0,0}^{(n)} x - s = c^{(n)} \\
x + v = u \\
x - z = l
$$

(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

$$
\tilde{d}(x^{(n)}) = \left( (\Theta_{0,0}^{(n)})^T S^{(n)} - \Theta_{0,0}^{(n)} + V^{(n)} - Z^{(n)} \right)^{-1} \Theta_{0,0}^{(n)}
$$

(17)

where $S^{(n)}$, $V^{(n)}$ and $Z^{(n)}$ are diagonal matrices with 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) and projects $\Theta_{0,0}^{(n)}$, which corresponds to $\tilde{d}(x^{(n)})$ given by (8). A geometrical intuition of how (17) modifies $\tilde{d}(x^{(n)})$ 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

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

(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 $\alpha$ 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

$$
H_{0,i}^{(n)} : E[F_i(x^{(n+1)})] \leq \alpha_i, \quad i = 1, \ldots, r.
$$

(19)

If the $r$ hypotheses $H_{0,i}^{(n)}$ 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**

1. Select initial point $x^{(0)} \in \mathcal{P}$
2. for $n = 0, 1, 2, \ldots$ do
3. Generate $\gamma^{(n)}$, set $p = 0$, and set $\varphi_i > \varphi^*_i$, for $i = 0, \ldots, r$
4. while one or more $\varphi_i > \varphi^*_i$ do
5. At each $x \in \gamma^{(n)}$, observe $r + 1$ metrics $F_i^\gamma y_{i=0}$
6. Insert the samples in $\{W_i^{(n)}\}_{i=0}^r$ and $\{y_i^{(n)}\}_{i=0}^r$
7. $p \leftarrow p + D$
8. Estimate $\Theta_{0,0}^{(n)}$, using (14)
9. Estimate $\varphi_i^y_{i=0}$ using (10), (11), and (12)
10. end while
11. Obtain $s^{(n)}$, $v^{(n)}$, and $z^{(n)}$ from (16)
12. Obtain $\tilde{d}(x^{(n)})$ using (17)
13. $x^{(n+1)} \leftarrow x^{(n)} + \alpha \tilde{d}(x^{(n)})$
14. Check the null-hypotheses $H_{0,i}^{(n)}$ defined in (19)
15. if one or more $H_{0,i}^{(n)}$ are accepted then
16. $x^{(n+1)} \leftarrow x^{(n)}$
17. end if
18. 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
evaluation [9]. Regarding the access network deployment and physical aspects the scenario consists of a hexagonal grid of 19 three-sectoral 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 intersite 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 \( \lambda \) 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 on the macrocell area, or in a smaller area around a picocell. The first case occurs with probability \( P_{\text{macro}} \), 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 \( \alpha, \delta \) and \( \phi \) 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 \( |f^* - F(x_i)| \) is smaller than a given value \( \epsilon \) for at least 10 consecutive samples. Different \( \epsilon \) values were used. For all the convergence evaluation experiments we selected the same initial configuration \( x(0) \in \mathcal{F} \). 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 (\( \alpha, \delta \) or \( \phi \)) was obtained by averaging the results of 200 independent simulation runs.

Figure 5 depicts the convergence time versus \( \alpha \), showing an interesting relation between accuracy and convergence time. On the one hand, smaller values of \( \alpha \) allow the system to operate closer to the optimum performance \( f^* \), but requiring more samples to attain convergence. Note, however, that increasing \( \alpha \) 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 \( d(x) \) decreases, the algorithm requires more samples to
### TABLE III
SIMULATION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network layout</td>
<td>19 macro eNBs, 57 directional macro cells, 500 m ISD, [2...12] pico cell per sector</td>
</tr>
<tr>
<td>System Bandwidth</td>
<td>10 MHz</td>
</tr>
<tr>
<td>Frame duration</td>
<td>Subframe 1 ms, Protected-subframe pattern 8 ms, Frame 10 ms</td>
</tr>
<tr>
<td>Transmit power</td>
<td>Micro cell 46 dBm, pico cell 30 dBm</td>
</tr>
<tr>
<td>Antenna Pattern (macro sector)</td>
<td>$A_H(\phi) = \min{12\left(\frac{\sin(\frac{\phi}{2})}{\phi}\right)^2, A_m}$, $A_m = 70$ degree, $A_h = 25$ dB</td>
</tr>
<tr>
<td>Antenna Pattern (pico)</td>
<td>Omnidirectional</td>
</tr>
<tr>
<td>Antenna gain</td>
<td>Macro: 14 dB; pico: 5 dB</td>
</tr>
<tr>
<td>Macro to UE path loss</td>
<td>$128.1 + 37.6 \cdot \log_{10}(R/\text{km})$ where $R$ is the macro eNB to UE distance</td>
</tr>
<tr>
<td>Pico to UE path loss</td>
<td>$149.7 + 36.7 \cdot \log_{10}(R/\text{km})$ where $R$ is the pico cell to UE distance</td>
</tr>
<tr>
<td>Shadow fading</td>
<td>Lognormal distribution with 10 dB standard deviation</td>
</tr>
<tr>
<td>Thermal Noise</td>
<td>-176 dB</td>
</tr>
<tr>
<td>Scheduling Algorithm</td>
<td>eICIC Proportional Fair (eICIC-PF) [35]</td>
</tr>
<tr>
<td>Traffic Model</td>
<td>File Transfer Protocol (FTP)</td>
</tr>
<tr>
<td>File size</td>
<td>0.5 Mbytes</td>
</tr>
<tr>
<td>$\Lambda$ (UE/s) (Offered traffic load) (Mbps)</td>
<td>0.5, 3.5, 10, 32.5 UE/s (2, 5, 130 Mbps)</td>
</tr>
<tr>
<td>Minimum distances</td>
<td>Macro - pico: 70 m; Macro - UE: 35 m; Pico - pico: 40 m; Pico - UE: 10 m</td>
</tr>
</tbody>
</table>

---

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

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

Accuracy and convergence rate. Increasing $\varphi^*$ implies faster convergence since fewer samples are required to estimate the search direction, but smaller $\varphi^*$ values are capable of operating closer to the optimum. Based on these results, the reference configuration selected for the following sections is $\alpha = 6.5 \times 10^{-5}$, $\delta = (0.03, 0.5)$, $\varphi^* = 7^\circ$ 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 ($\gamma$ and $\phi$) 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 5th 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 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 $\text{d}(x^{(0)})$ requires a higher amount of samples at each iteration.

Finally, Figure 7 shows the convergence time as a function of $\varphi^*$. For this parameter, there is a clear tradeoff between
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, $P_{\text{pico}} = 2/3$), (130 Mbps, $P_{\text{pico}} = 2/3$), (20 Mbps, $P_{\text{pico}} = 7/8$), and (20 Mbps, $P_{\text{pico}} = 7/8$). These values were obtained by an exhaustive search over a set of discretized values of $P$. 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 $\gamma$ 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 to 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 $d_{\text{opt}}(x)$. The objective is to quantify the effect of using $d_{\text{opt}}(x)$ in terms of convergence rate. For RC-RSM we use two $\alpha$ 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.
Fig. 10. 5th percentile throughput as a function of offered traffic load for the proposed benchmarks.

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 5th 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 $x^{(0)}$. The faster convergence of SR-RSM compared to RC-RSM illustrates the convenience of the modified search direction $\tilde{d}(x)$ instead of $d(x)$. 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 5th or the 50th 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 5th and the 50th percentile throughputs 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 5th or the 50th percentile throughput. Figure 14 shows the SR-RSM curves and the PF-ABS operation points for each scenario.

Note: 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.

REFERENCES


Data-Driven Configuration of Interference Coordination Parameters in HetNets
Jose A. Ayala-Romero ©, Juan J. Alcaraz ©, and Javier Vales-Alonso ©

Abstract—The heterogeneous networks (HetNets) performance is highly dependent on the interference coordination among cells since both macrocells and small cells share the same spectrum. This paper focuses on the configuration of eICIC parameters: almost blank subframe ratio and cell range expansion bias defined for LTE-A. We propose an online learning mechanism based on a big data-driven framework and multiarmed bandit algorithms that retrieves data from the network to learn efficient configurations of eICIC parameters without requiring any previous knowledge about the network (e.g., traffic load, topology, scheduling algorithm). Our numerical results show that our approach attains a significant improvement with respect to the state of the art of online learning algorithms in networks under stationary and variable conditions (e.g., number of small cells, traffic load).

Index Terms—Interference coordination, heterogeneous networks, data-driven, model-free, online learning.

1. INTRODUCTION

The amount of data produced in cellular network by mobile phones and other smart devices is increasing exponentially [1]. This creates an opportunity for network operators to extract relevant information using big data techniques and apply algorithms that exploit this information for real-time decision making in a wide range of applications. Specifically, we address the interference coordination management problem in heterogeneous networks (HetNets) using a data-driven approach. Thus, we propose to use the data retrieved from the network in order to learn efficient configurations of the interference coordination parameters and, attending to the dynamic nature of the network, perform a real-time adjustment of the interference management parameters according to variations in network conditions.

We consider a typical HetNet scenario with small cells overlapping the macro cell coverage area. Since all cells share the same spectrum, the interference coordination management is essential, specially where the small cell deployment density increases. We apply our proposal to the enhanced Inter Cell Interference Coordination (eICIC) technique defined by 3rd Generation Partnership Project (3GPP) for Long Term Evolution Advanced (LTE-A) Network. The eICIC parameters are Cell Range Expansion (CRE) bias and Almost Blank Subframe (ABS) ratio.

The eICIC technique performs a resource allocation among macro cells (macro eNB in 3GPP terminology) and pico cells (pico eNBs) in the time domain. The optimal configuration of eICIC parameters depends on the dynamically varying network conditions (traffic load, user demands on positions, number of active pico eNBs, etc) [2]. However, most previous works consider a static situation of the network [3]–[11]. To overcome this limitation, we formulate the problem as an online learning problem where the proposed mechanism configures the eICIC parameters in real-time, during the operation of the network.

Another limitation of previous works is that their approaches rely on mathematical models of the network [3], [6], [12]–[14]. In general, even the most complex network models may not comprise all the relevant aspects of a real operating network. In contrast to previous works, our approach is data-driven instead of model-driven. Thus, it operates without making any assumption about the network.

As a part of our proposal, we propose a novel Multi-armed bandit (MAB) algorithm for a balanced exploration/exploitation decision making in eICIC parameter configuration. The objective of MAB algorithms is to maximize an expected reward over time, given a set of actions whose individual rewards are initially unknown and possibly random (as in our case). Our proposal is capable of adapting to variable network conditions, continuously adjusting the eICIC parameters to configurations that are efficient in the current conditions.

Finally, we evaluate our proposal against two families of benchmark online algorithms: stochastic gradient ascent (SGA) algorithms and MAB algorithms. We consider two settings: stationary and variable conditions in the network.

The rest of the paper is organized as follows. In Section II the related work and contribution summary are given. In Section III we describe the proposed big data framework and detail the eICIC configuration parameters. The network model and the formulation of the problem are given in Section IV. In Section V we describe the proposed mechanism. The benchmark algorithms are described in Section VI. Finally, the numerical results are given in Section VII and the conclusions are summarized in Section VIII.

II. RELATED WORK AND CONTRIBUTION

A. Related Work

With recent advances in big data analytics, data-based cellular network optimization has attracted the attention of the scientific community [1], [2], [19]. The authors in [1] propose a frame-
work for big-data driven mobile network optimization and also suggest that it can be used for interference management.

Some works like [4], [8], [18] address this problem using learning algorithms. The authors in [8] use a reinforcement learning (RL) approach, specifically a fuzzy Q-learning algorithm, to learn the optimal ABS ratio considering a fixed CRE bias. In [4], Q-learning algorithms are also proposed for ABS ratio and CRE bias learning. However, in [4], [8] the learning algorithms are applied to static network situation, i.e., constant traffic intensity and UEs with fixed locations. In contrast, in [18] and our work, the variable nature of the network is considered. The main issue with RL approaches is that they require offline training before operating in the network [18]. This training is usually carried out in a network simulator introducing the additional requirement of reproducing the network in a simulator model. If the simulator model is not accurate the RL policies may be inefficient in practice. In contrast, our proposal operates without offline training or any previous knowledge about the network.

Most related works address a static network situation [3]–[11], not considering explicitly the real-time adaption to a dynamic network conditions. Although some works consider the dynamic nature of the network [13], [15]–[17], they differ from ours in the following aspects: First, the authors of [15]–[17] propose a heuristic approach which is scenario dependent. For example, in [15] they assume the use of Proportional Fair (PF) scheduling in the HetNet using the PF metric as a basic element of the heuristic. Second, the approaches proposed in [13], [15], [16] only configure the ABS ratio ignoring that both ABS ratio and CRE bias have a joint impact in the performance metric [17]. In contrast, our proposal is able to jointly configure these two parameters regardless of the particular scenario, only using data retrieved from the network.

Let us discuss the difference between model-driven schemes and our data-driven, model-free approach. Model-driven approaches (e.g., [3], [6], [12]–[14]) need to collect key parameters from the network (such as the location of all UEs or channel gains) to feed the model which necessarily contains assumptions and/or simplifications of the real network making it suitable for an optimization algorithm. However, the optimal configuration for the model might not necessarily be the optimal configuration of the real network due to the simplification and/or assumptions in the network model. In contrast, a data-driven approach relies on direct observations of the network performance obtained from the real network (without assumption or simplifications). This allows the decision algorithm to explore configurations aiming at finding the most efficient ones. For example, the scheme in [12] aims at optimizing a weighted sum of the logarithms of the user throughputs. The algorithm parameters are the interference graph of the network and the data rates that each user would obtain if attached to the nearest macro station (rMF m) of to the nearest pico station (rPF p). The associated signaling overhead of this scheme is higher than ours, but the main drawback is the implicit assumption of considering these data rates, (rMF m, rPF p) independent of the eICIC configuration. Note that this assumption is a simplification/approximation, since the rate at each UE is determined by the SINR, which, in turn, depends on the eICIC configuration, and therefore rMF m and rPF p cannot be accurately known a priori. This assumption, among others, is useful in setting up a convex problem, but may imply that the optimal configuration for this model does not match the optimal configuration of the real system, especially if the desired performance metric is not captured by the objective function. In the numerical results of that paper it is shown that the 5th percentile throughput of some fixed configurations is practically equal to the proposed algorithm, which suggests that a search over the configuration set (as the one proposed in our paper) would eventually find a configuration at least as good as the one found by this algorithm.

Some works like [20], [21] address the problem using stochastic geometry. This approach models the location of macro eNBs as a homogeneous poisson point process (PPP). It implies that the location of each macro eNB does not depend on others. In a real network the deployment of each macro eNB is generally planned and depends on the position of the surrounding macro eNB, the terrain, attenuation profile, etc. The authors in [20] also assume Round Robin as a scheduling policy, not appropriate for HetNets. In contrast, our mechanism relies on real data from the operating network avoiding the simplifications introduced by mathematical models.

Finally, our proposal also differs from previous works in terms of computation and signaling overhead, not imposing a noticeable burden in any of them. For example, the algorithm in [12] requires information about network topology and also computes at each iteration the following data: the data-rate achievable by each UE if it were connected to best macro, the data-rate achievable by each UE if it were connected to best pico in ABS and non-ABS subframe and the mean throughput of each UE. In a similar way, [3] and [22] require the interference graph of the network to operate.

We provide a detailed comparison between our work and related works in Tables I and II.

### Table I

<table>
<thead>
<tr>
<th>Controlled Parameters</th>
<th>[4]</th>
<th>[15], [16]</th>
<th>[13]</th>
<th>[8]</th>
<th>[9], [10]</th>
<th>[5], [12]</th>
<th>[11]</th>
<th>[19]</th>
<th>[14]</th>
<th>[18]</th>
<th>DonC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Operation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Coordinated Solution</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Black box approach</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Approach: L = Learning, O = Optimization, H = heuristic.

### Table II

<table>
<thead>
<tr>
<th>Simulation Model Comparison in Related Works</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TABLE II</strong></td>
</tr>
<tr>
<td><strong>WORKS</strong></td>
</tr>
<tr>
<td><strong>Static</strong></td>
</tr>
<tr>
<td>[3], [4], [7]</td>
</tr>
<tr>
<td><strong>Traffic Model</strong></td>
</tr>
<tr>
<td>[3], [4], [7]</td>
</tr>
<tr>
<td><strong>Dynamic</strong></td>
</tr>
<tr>
<td>[12]–[15], [17], [18], Ours</td>
</tr>
</tbody>
</table>

We provide a detailed comparison between our work and related works in Tables I and II.

### B. Contribution

The first contribution of this paper is to present a specific application of the general idea of data-driven network management, described in [1], [23], as a proof-of-concept. We show how this approach can be applied to the dynamic configuration...
of eICIC parameters in HetNets in our proposed mechanism: the Data-Driven eICIC Configurator (DDeC).

Our proposal finds efficient configurations of the eICIC parameters using only data retrieved from the network. This data-driven nature provides two main benefits to DDeC with respect to previous approaches: first, DDeC can operate without any previous knowledge of the underlying network (e.g., topology, scheduling algorithms); second, DDeC can optimize any measurable performance metric, even those that are analytically intractable such as the 5th percentile throughput.

DDeC is based on a novel online learning algorithm of the Multi-Armed Bandit (MAB) type. Compared to previous MAB algorithms, our proposal comprises the following novelties: 1) it uses a local exploration strategy which notably increases the algorithm's efficiency, 2) it uses only a reduced set of actions on each learning/decision stage, 3) it can operate under non-stationary network conditions by means of a novel module that detects if the optimal action changes. This addresses a major challenge for online learning algorithms. In consequence, our numerical experiments show that DDeC outperforms previous proposals. Besides, it does not introduce additional computational complexity compared to previous approaches.

III. BIG DATA FRAMEWORK AND APPLICATION SCENARIO

A. Big Data Framework

Our proposal is built upon the big data-driven framework proposed in [1] and depicted in Fig. 1. This framework integrates big data analytics and network optimization with the objective of improving the user quality of service. It can be applied to resource management in HetNets such as network planning optimization, energy saving, etc. However, our work is, to the best of our knowledge, the first to apply this framework to interference coordination in HetNets.

The framework, adapted to our eICIC functionality, consists on the following data flow: Data Collector and Processor (DCP) collects raw data from the network and processes it to obtain suitable data to feed the DDeC (Data-Driven eICIC Configurator). According to this processed data, DDeC selects the eICIC configuration for the operating network. This new eICIC configuration affects the raw data collected henceforth which is used as a feedback. The successive iterations of this process allows DDeC to find efficient configurations.

Regarding the data collection, the raw data is computed at eNBs and sent to the DCP to be processed. Thus, DCP obtains a network statistic or performance sample from the whole network.

In our scenario, the time varying nature of the network is a relevant aspect, since it affects, in general, to the efficiency of the eICIC configurations. Therefore, DDeC incorporates the capability of detecting statistical variations in the network conditions, in real time, to enable a fast adaptation of the interference management configuration.

B. Interference Management in LTE-A: eICIC

The eICIC is an interference management technique defined by 3GPP in Release 10 (LTE-A) [24] for HetNet environments. To prevent inter-cell interference, eICIC allows macro eNBs and pico eNBs to use radio resources in different time periods (subframes). The main features of eICIC are: Cell Range Expansion (CRE) and Almost Blank Subframe (ABS).

Pico eNBs are intended to enhance the spatial frequency reuse of the network. However, since the transmission power of macro eNBs is higher than the transmission power of pico eNBs, a UE close to a pico eNB could associate with the macro eNB with high probability, losing the benefits provided by pico eNBs. It occurs because LTE networks implement association based on received signal reference power (RSRP), i.e., UEs associate with the highest reference signal that every cell broadcasts. Therefore, it leads to the underutilization of the pico eNBs and the overload of the macro eNBs. To cope with that problem, CRE allows the UE to associate to a pico eNB even when its RSRP is lower than the RSRP from the macro eNB. For that purpose, the UEs add the CRE bias to pico eNBs extending the pico eNBs footprint. Thus, the UE associates with the eNB with maximum (corrected) RSRP. Nevertheless, the UEs located at pico eNB extended region experience a poor Signal to Interference and Noise Ratio (SINR) due to the high power received from the macro eNB.

To alleviate this problem ABS is introduced. It allows the macro eNBs to mute all data symbols in certain subframes referred to as Almost Blank Subframes. In these protected subframes the UEs with poor channel conditions can boost its SINR due to the absence of macro interference in data symbols. Although a LTE frame comprises 10 subframes, the ABS patterns has a periodicity of 8 subframes. Thus, it is necessary to configure the ratio of protected subframes over conventional subframes (0/8, ..., 8/8). We assume ABS ratio mixtures are also possible (e.g., 2/8, 3/8 with probability 0.5 each) obtaining another ABS ratio value (e.g., 2.5/8). As a result, the ABS ratio can take values in [0, 1].

As in [20], [25], [26], we consider synchronized muting and common CRE bias value.

IV. PROBLEM FORMULATION

A. Network Performance Metric

Consider a network operator who wants to optimize a certain performance metric $P$ which depends on the interference coordination parameters of the network. In our scenario, these parameters are the ABS ratio ($\gamma$) and CRE bias ($\phi$). Note that,
for any configuration \( x = (\gamma, \phi)^T \) and under stationary network conditions (i.e., traffic intensity, number of active pico eNBs), the value of \( F(x) \) is a random variable due to the stochastic nature of the network (i.e., variable UE positions, stochastic shadow fading).

The performance metric \( F \) can be related, for example, to the physical layer (signal to interference and noise ratio, SINR) or to upper layers (data throughput). Since a mathematical characterization of \( F(x) \) for a real operating network without simplifications is unavailable in general, our approach aims at finding efficient configuration based on performance observation from the network. Thus, \( DCP \) (Fig. 1) is responsible for processing the raw data collected from the network in order to obtain an observation of the random variable \( F(x) \) for the configuration \( x \). We consider \( DCP \) as a black box and its operation is out of the scope of this paper.

Although \( DDeC \) operates independently of the selected performance metric \( F \), in our case, we consider the 5th percentile throughput over the downlink to be the performance metric. Specifically, \( F(x) \) is the MLE estimator (and thus a random variable) of the 5th percentile throughput using the configuration \( x \). This metric is proposed by the 3GPP [27] to evaluate the performance of LTE networks using non-full buffer traffic model. In addition, multiple related works [12], [15], [25] use this metric as well. We define a throughput sample as the quotient between the size of a downloaded file and the time needed to download it [27]. The 5th percentile throughput is defined as the value below which 5 percent of the throughput samples may be found. Thus, given a set of throughput samples retrieved under configuration \( x \), we can obtain a sample of the 5th percentile throughput which is a random variable denoted by \( F(x) \). The set of values \( E[F(x)] \) for all \( x \) is referred to as the system response. Fig. 3 shows an example of system response for specific network conditions. Note that when the network conditions varies, the system response generally changes.

Regarding the big data-driven framework depicted in Fig. 1, in our case, the raw data retrieved by \( DCP \) are throughput samples obtained by UEs. Processed data refers to 5th percentile throughput samples computed by \( DCP \) from throughput samples.

### B. Online Problem Formulation

In this section, we formulate the interference management problem as an online learning problem. In this type of problem, in contrast to offline approaches, the algorithm has to select configurations and sample their performances from the network as feedback. Since changing the configuration actually affects the network performance, we face the challenge of selecting configurations allowing us to infer the system response, while avoiding poor performing ones. In addition, in our case, we must face the non-stationarity of the network conditions, which causes variations in the system response.

The performance samples are obtained in countable time stages \( k = 1, 2, \ldots \). At each stage \( k \) a configuration \( x_k \) is selected and a performance sample \( y_k = F(x_k) \) is obtained. The next configuration \( x_{k+1} \) is selected according to the knowledge obtained by past samples. All configurations \( x_0 \ldots x_k \) should be selected from the set \( \mathcal{P} = \{ x = (\gamma, \phi)^T : 0 \leq \gamma \leq 1, 0 \leq \phi \leq \phi_{\text{max}} \} \).

![Fig. 2. Detailed system scheme with DDeC submodules.](image)

The pseudo-regret up to stage \( k \) of an algorithm is defined by:

\[
     kE[F(x^*)] - \sum_{i=1}^{k} E[F(x_i)]
\]

where

\[
     x^* = \arg \max_{x \in \mathcal{P}} E[F(x)].
\]

The proposed mechanisms are aimed at minimizing the pseudo-regret (hereinafter referred to as regret) which is the most used metric in online settings [28], [29]. Although the problem is formulated over the continuous set \( \mathcal{P} \) of configurations, our proposal addresses it only selecting actions from a discrete set \( \mathcal{A} \) as we detail in the next section.

Note that we have defined two performance metrics: first, the 5th percentile throughput \( (F) \) which is the performance metric that our proposal uses and is obtained directly from the network data. Second, the regret, which shows how good is an algorithm in an online setting. The regret is computed from samples of 5th percentile throughput. Specifically, it captures the accumulated loss of not selecting the optimal configuration \( F(x^*) \) at each sample, and therefore shows how the algorithm behaves with respect of the optimal performance. It is worth to remark that, since the optimal \( x^* \) is unknown in advance the regret cannot be computed in a real network. However, it can be estimated by exhaustive search in numerical experiments. We detail in Section VII-B how we address regret computation.

Note that the optimal configuration \( x^* \) can vary over time making it more challenging for the algorithms to select efficient configurations. Nevertheless, since low regret implies that the loss associated to a suboptimal configuration is also low, the aim of an online algorithm is to minimize this metric over time. For that purpose, there is a broad literature about SGA and MAB algorithms addressing regret minimization in online learning problems [28], [30]–[35]. The following section presents a novel mechanism showing a significant improvement with respect to SGA and MAB state of the art algorithms.

V. MECHANISM DESCRIPTION

This section describes \( DDeC \) depicted in Fig. 1 and further detailed in Fig. 2. This module sends at the beginning of each stage \( k \) the configuration \( x_k \) to be used in the network. Then, the network operates with this configuration for the whole duration of the stage \( k \) sending the raw data (in our case throughput samples) to \( DCP \) which processes it in order to obtain a performance sample \( y_k = F(x_k) \). This performance sample is sent to \( DDeC \) which uses it to select the next configuration \( x_{k+1} \).

Fig. 3 shows the expectation of \( F(x) \) for all \( x \in \mathcal{A} \) (system response). We can verify in this figure that the expected
performance of every action is correlated with the expected performance of the actions of its surrounding, which means that the region of the optimum configuration can be reached from any other configuration following a sequence of actions with strictly increasing performance. This characteristic was verified in a wide range of scenarios with different configurations (traffic load, number of pico eNBs, etc.) by means of extensive simulation and is in line with the results obtained in [20]. Our proposal exploits this structure in the system response making local explorations and selecting the best local actions in order to determine the next set of actions for local exploration. Thus, we can approach the optimum action quickly and without exploring all the available actions in $\mathbb{A}$.

DDDeC has two objectives: first, to find efficient actions by exploiting the above described structure of the system response, and second, to detect changes in the network conditions affecting this response. In order to accomplish these objectives, it is divided into three submodules: the Action Availability Selector (AAS), the Multi-Armed Bandit with limited action availability (MAB-LAA) and the Network Change Detector (NCD). These three submodules can be seen as three black boxes operating transparently from the others and only using the input/output data detailed in Fig. 2. The tasks of each submodule are summarized as follows:

- **AAS** generates at each stage a discrete set $\mathbb{A}_k$ of configurations $x$ (also called actions) that can be selected by our mechanism.
- **MAB-LAA** selects which configuration $x$ to use at each stage from the set of available actions $\mathbb{A}_k$ given by AAS.
- Finally, **NCD** tracks the values of the observations in order to notify to other submodules when it detects a variation on the network conditions.

### A. Action Availability Selector (AAS)

The aim of AAS submodule is selecting the subset $\mathbb{A}_k \subset \mathbb{A}$ of actions that can be chosen by our proposal at each stage $k$, from the set $\mathbb{A}$ of all available actions. The operation of AAS comprises two periods: **search period** and **exploitation period**.

The purpose of the search period is to rapidly approach the region of the optimal action $\mathbb{x}^*$ from any initial action $\mathbb{x}_0$. The exploitation period performs a fine-tuned action exploration in the region of $\mathbb{x}^*$ in order to find actions with better performance.

---

**Fig. 4.** Example of algorithm evolution from AAS submodule point of view. In the search period, the set $\mathbb{A}^{i+1}$ is created from the best local action of iteration $i$ (highlighted in green). Then for $\mathbb{A}^{i+2}$, created from the best local action of iteration $i+1$ (highlighted in red). The red action is the best local action during the next $\alpha$ iterations which indicates the beginning of the exploitation period. The actions remarked with crosses ($\mathbb{A}^{\text{exp}}$) are the available action during the whole exploitation period.

The search period is divided in iterations. We define an iteration as a set of $\tau$ stages where the set of available actions does not change. Let $\mathcal{I} = \{i : k < i \leq k + \tau\}$ be the set of $\tau$ consecutive stages belonging to iteration $i$. For each iteration, AAS selects a set of available actions $\mathbb{A}^k$ such that $\mathbb{A}_k = \mathbb{A}^k$ for all $k \in \mathcal{I}$. Fig. 4 depicts an example of $\mathbb{A}$ (all possible actions with $x$ 2-dimensional) and the evolution of three consecutive iterations of search period. Note that, for each iteration $i$, all actions in the set $\mathbb{A}^k$ wrap around the so-called central action ($\mathbb{x}_c^k$). Thus, at first iteration ($i = 0$), we choose an initial central action $\mathbb{x}_c^0$ and 8 additional actions wrapping around $\mathbb{x}_c^0$ to define the set $\mathbb{A}^0$. The set $\mathbb{A}^1$ is selected by AAS which sends it to MAB-LAA at each stage $k$.

AAS stores in $\mathbb{Y}^k_x$ the number of times that each action $x \in \mathbb{A}^k$ has been selected during the iteration $i$. At the end of $i$-th iteration, MAB-LAA determines the best local action, given by

$$\hat{x}_i = \arg \max_{x \in \mathbb{A}^k} \mathbb{Y}^k_x,$$  \hspace{1cm} (3)

Then, $\hat{x}_i$ is used as the central action for next iteration ($\mathbb{x}_c^{i+1} = \hat{x}_i$). The configuration of parameter $\tau$ will be discussed later in Section VII-F.

When the best local action of iteration $i$ matches the current central action ($\hat{x}_i = x_i$), the set $\mathbb{A}^{i+1}$ will contain the same elements of $\mathbb{A}^i$. The exploitation period starts when the set of available actions does not change during $m \geq 2$ consecutive iterations, that is, $\hat{x}_i = x_i^{i-1} = \ldots = x_i^{i-m+1}$ (illustrated in Fig. 4 at iteration $i+2$ and in Algorithm 1 from line 16 onwards). Then, a new set of available actions $\mathbb{A}^{\text{exp}}$ is used, containing configurations closer to the last best local action. Note that $\mathbb{A}_k = \mathbb{A}^{\text{exp}}$ during the whole exploitation period.

The exploitation period benefit is twofold: (i) the set $\mathbb{A}^{\text{exp}}$ is generated with actions closer to $x_c$, which is the action with the highest estimated performance so far. This aims at exploring the optimal region in depth to find actions with higher performance than $x_c$, (ii) one of the main drawbacks of SGA algorithms is that they deviate from the optimal region after approaching that due to stochastic nature of $F(x)$. In contrast, in the exploitation period of DDDeC the set of available actions $\mathbb{A}^{\text{exp}}$ remains fixed.
Algorithm 1: AAS framework.

1: Input parameters: $x_i^0$, $\tau$
2: Initialize: $k = 0$
3: for $i = 0, 1, 2, \ldots$ do
4: Generate the set $\mathcal{A}_i$ from $x_i^i$
5: Generate the set $\mathcal{D}_i = \{i : k \leq i \leq k + \tau\}$
6: while $k \in \mathcal{D}_i$ do
7: $\mathcal{A}_k = \mathcal{A}_i$
8: Send $\mathcal{A}_k$ to MAB-LAA
9: $x_k$ is received and sent to the HetNet
10: $y_k$ is received and sent to the MAB-LAA
11: Update $\mathcal{Y}_x^i$ with $x_k$ and $y_k$
12: $k = k + 1$
13: end while
14: Compute $x^i = \arg\max_{x \in \mathcal{A}_i} \mathcal{Y}_x^i$
15: $x^{i+1} = x^i$
16: if $x^{i+1} = x^{i-1} = \ldots = x^{i-m+2}$ then
17: Generate the set $\mathcal{A}^{exp}$ from $x^{i+1}$
18: while Notify from NCD is not received do
19: $\mathcal{A}_k = \mathcal{A}^{exp}$
20: Send $\mathcal{A}_k$ to MAB-LAA
21: $x_k$ is received and sent to the HetNet
22: $y_k$ is received and sent to the MAB-LAA
23: Send $x_k$ and $y_k$ to NCD
24: $k = k + 1$
25: end while
26: end if
27: end for

until receiving a NCD notification, avoiding the aforementioned deviation. The operation of the AAS submodule is summarized in Algorithm 1.

B. Multi-Armed Bandit with Limited Action Availability (MAB-LAA)

This submodule selects the action $x_k$ to use at each stage $k$ from the set $\mathcal{A}_k$ which is provided by AAS submodule at each stage. Algorithm 2 summarizes the operation of MAB-LAA submodule. The input parameter $\bar{n}$ and how the actions are selected (line 5) are determined by the selected MAB policy.

Although any MAB policy can be used in the MAB-LAA framework, we use the Thompson sampling normal policy [30] (described in Section VI-B) because it obtains better performance compared to other MAB policies in our numerical evaluations.

C. Network Change Detector (NCD)

NCD submodule is responsible for detecting changes in the system response by using the samples of $F(x)$. This detection is only active in the exploitation period of AAS. When a change in the system response is detected, the submodules AAS and MAB-LAA are notified by NCD (Fig. 2) and, as a consequence, AAS switches to search period and MAB-LAA resets its knowledge about the random variable $F(x)$ for all $x$. Then, the central action of the first iteration of AAS is set to the best local action in the exploitation period.

The notification mechanism of NCD is based on a hypothesis test which checks if there is a statistically significant change in the samples of $F(x)$ assuming normal distribution of this random variable. Let $\mathcal{R}^x$ be the reference set composed by the first $\nu$ samples obtained in the exploitation period using the configuration $x$. Let $\mathcal{T}^x$ be the temporal set composed by the last $\nu$ samples obtained in the exploitation period using the configuration $x$. Note that there is a reference set and a temporal set for each available action in the exploitation period. Let us denote by $\mu_x^c$ and $\mu_x^t$ the average of the sets $\mathcal{R}_x$ and $\mathcal{T}_x$, respectively. A hypothesis test is carried out for each $x \in \mathcal{A}^{exp}$ considering the following null hypothesis:

$$H_0 : \mu_x^c = \mu_x^t \quad \forall x \in \mathcal{A}^{exp}$$

(4)

Note that the hypothesis test for action $x$ is performed every time that $\nu$ new observations of action $x$ are taken. In order to minimize the loss produced by a false positive in the hypothesis test (false detection of a meaningful change in the network conditions), we consider that the system response has changed if 5 consecutive tests with significance level equals to 0.01 fail. The operation of NCD module is detailed in Algorithm 3. Table III summarizes the most relevant parameters of the proposal.

D. Signaling and Computation Overhead

As we detail at the beginning of this section, the DDeC sends a configuration $x_k$ to the network at each stage $k$, that is, the stage duration defines the time period between two consecutive configurations. Therefore, this is a design decision which has to be properly tuned according to the change rate of network conditions.

We define a stage as the time needed to collect enough data from the network to obtain one 5th percentile throughput sample. The duration of one stage can be variable depending on the amount of data generated by the network. Considering that the number of throughput samples needed by DCP to compute one 5th percentile throughput sample with statistic guarantee is fixed, the duration of one stage is a design decision depending on:

- The time required to obtain a throughput sample. Since the throughput is defined as the length of the packet divided by the time required to download it, we have to define the throughput packet length. For example, given 2 Mbytes of downloaded data, we obtain either 4 or 20 throughput samples for a throughput packet length of 0.5 Mbytes or 100 Kbytes, respectively.
### TABLE III

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_k$</td>
<td>Action or configuration selected at stage $k$</td>
</tr>
<tr>
<td>$y_k$</td>
<td>Performance sample ($5^{th}$ percentile throughput) obtained from stage $k$</td>
</tr>
<tr>
<td>$\mathcal{A}$</td>
<td>Set of all possible actions or configurations</td>
</tr>
<tr>
<td>$r$</td>
<td>Number of stages of each iteration</td>
</tr>
<tr>
<td>$\mathcal{A}_i$</td>
<td>Set of actions of iteration $i$</td>
</tr>
<tr>
<td>$T$</td>
<td>Set of $r$ stages corresponding to iteration $i$</td>
</tr>
<tr>
<td>$s_i$</td>
<td>Central action of iteration $i$</td>
</tr>
<tr>
<td>$s^*$</td>
<td>Best local action of iteration $i$</td>
</tr>
<tr>
<td>$\mathcal{A}^{\text{exp}}$</td>
<td>Set of actions of exploitation period</td>
</tr>
<tr>
<td>$\mathcal{A}_k$</td>
<td>Set of available actions for stage $k$</td>
</tr>
<tr>
<td>$\bar{n}$</td>
<td>Minimum number of times that any action has to be selected in the algorithm initialization</td>
</tr>
<tr>
<td>$\nu$</td>
<td>Size of $\mathcal{D}_x^\nu$ and $\mathcal{D}_s^\nu$</td>
</tr>
<tr>
<td>$\mathcal{D}_x^\nu$</td>
<td>Reference set, first $\nu$ performance samples of action $x$ in the exploitation period</td>
</tr>
<tr>
<td>$\mathcal{D}_s^\nu$</td>
<td>Temporal set, last $\nu$ performance samples of action $x$ in the exploitation period</td>
</tr>
<tr>
<td>$\mu_x^\nu, \mu_s^\nu$</td>
<td>Average of the sets $\mathcal{D}_x^\nu$ and $\mathcal{D}_s^\nu$, respectively</td>
</tr>
</tbody>
</table>

#### Algorithm 3: NCD operation.

1. **Initial Variables:** $\mathcal{R}^1 = \emptyset$, $\mathcal{T}^1 = \emptyset$, counts $= 0$
2. **for** each stage $k$ **do**
3.     **if** Exploitation period active **then**
4.         **Receive** $x_k \in \mathcal{A}^{\text{exp}}$ and $y_k$
5.         **if** length($\mathcal{R}^k_x$) $< \nu$ **then**
6.             **Append** $y_k$ to $\mathcal{R}^k_x$
7.         **else**
8.             **Append** $y_k$ to $\mathcal{T}^k_x$
9.         **if** length($\mathcal{T}^k_x$) $= \nu$ **then**
10.             **if** HypothesisTest($\mathcal{R}^k_x$, $\mathcal{T}^k_x$) fails **then**
11.                 counts $= \text{counts} + 1$
12.             **else**
13.                 counts $= 0$
14.             **end if**
15.         **end if**
16.         **Empty** the set $\mathcal{T}^k_x$
17. **else**
18.         **if** counts $= 5$ **then**
19.             **Send Notification** to NCD and MAB-AAS
20.         **Empty** the sets $\mathcal{R}^k_x$ and $\mathcal{T}^k_x \forall x \in \mathcal{A}^{\text{exp}}$
21. **end if**
22. **end if**
23. **end for**

- The traffic intensity. Considering that the network is not saturated, the more UEs in the network, the more throughput samples we will get per second.
- The size of the network. Similarly to the previous item, a larger network contains more UEs which will provide more throughput samples per second.

Thus, adjusting the throughput packet length as a function of the current traffic intensity and the network size, the algorithm can operate at the desired adaptation rate and signaling overhead.

Regarding the size of the network, it is divided into clusters of several sectors sharing the same ABS pattern (synchronized muting\(^1\)). It implies that we only need one control per cluster of macro eNBs, i.e., we need as many algorithm instances as clusters we have. These clusters have to be defined by the network operator, grouping together all macro eNBs with homogeneous (similar) traffic profile, e.g., the city center, the outskirts of the city, etc. Note that the input of the algorithm is fixed regardless of the size of the network. Given a fixed input, the number of operation is always bounded ($O(1)$). In terms of storage overhead, our proposal only needs to store three elements for each configuration in $\mathcal{A}^i$, i.e., a total of $3 \cdot |\mathcal{A}^i|$ scalars, where the cardinality of $\mathcal{A}^i$ is fixed for all $i$.

### VI. BENCHMARK ALGORITHMS

In this section we present the benchmark algorithms used for a comparative evaluation of our proposal. We consider only algorithms that, like ours, use an online approach.

When evaluating a benchmark, we replace DDeC module (Figs. 1 and 2) with the corresponding benchmark algorithm. Note that the input/output of benchmark algorithms match with that of DDeC.

#### A. Stochastic Gradient Ascent (SGA) Algorithms

The operation of a SGA algorithm can be summarized in three steps: 1) Take performance samples around the current action, 2) compute an estimation of the gradient using these samples, 3) move to a new location following the direction of the gradient estimation using the projector operator $\Pi_{\mathcal{P}}$ over the set $\mathcal{P}$ (i.e., $\Pi_{\mathcal{P}}(x)$ returns the configuration in $\mathcal{P}$ closest to $x$).

Algorithm 4 shows the general framework for SGA algorithms. These algorithms consider $x$ to be a continuous variable confined in the set $\mathcal{P}$. Note that the operation of the framework is divided in iterations denoted by $t$. Each iteration is composed of $|\mathcal{S}^{(t)}|$ stages where $\mathcal{S}^{(t)}$ denotes the set of actions to sample at iteration $t$. The differences among the SGA algorithms presented below are basically two: how the set $\mathcal{S}^{(t)}$ is generated and its size, and how the estimation of the gradient is computed.

**One Sample Gradient (OSG)\(^{[31]}\)** uses an approximation of the gradient that is computed using a single random performance sample $\mathcal{S}^{(t)} = \{x^{(t)} + \delta u^{(t)}\}$ of the metric $F$, where $u^{(t)}$ is a random unit vector and $\delta$ is a constant determining how far of \(^1\)Some previous works [15], [16] have considered unsynchronized muting, i.e., that the ABS pattern can be different on each macro sector, leading to an uncertain interference profile, and hence to a more complex resource allocation problem. Since 3GPP has shown [26] that the use of synchronized muting implies a significant performance gain over unsynchronized one, we considered synchronized operation in our simulation scenario.
1: Input parameters: Learning rates $\eta_t$, initial point $x_0$
2: for $t = 1, 2, \ldots$ do
3:   Generate the set of configurations to sample $S^t$
4:   Obtain the performance samples of the metric $F$ at $S^t$
5:   Compute the gradient $\tilde{g}^t$
6:   Update $x^{(t+1)} = \Pi_F(x^{(t)} + \eta_t \tilde{g}^t)$
end for

The observation is taken. Then, the gradient is estimated by

$$\tilde{g}^t = \frac{\partial F(x^{(t)} + \delta u^{(t)})}{\partial u} u^{(t)}.$$  \hspace{1cm} (5)

**Multi-Sample Gradient with 2 samples (MSG2) [32]** takes two performance samples $S^t = \{x^{(t)} \pm \delta u^{(t)}\}$ and the gradient is estimated by

$$\tilde{g}^t = \frac{1}{2} (F(x^{(t)} + \delta u^{(t)}) - F(x^{(t)} - \delta u^{(t)})) u^{(t)}.$$  \hspace{1cm} (6)

**Multi-Sample Gradient with 3 samples (MSG3) [32]** considers $S^t = \{x^{(t)}, x^{(t)} + \delta e_1, x^{(t)} + \delta e_2\}$ where $e_i$ are the unit coordinate axes. The gradient is estimated using

$$\tilde{g}^t = \frac{1}{3} \sum_{i=1}^{3} (F(x^{(t)} + \delta e_i) - F(x^{(t)})) e_i.$$  \hspace{1cm} (7)

Response Surface Methodology (RSM) [33] considers $S^t = \{x^{(t)}, x^{(t)} + (\delta_1, \delta_2)^T, x^{(t)} + (\delta_1, -\delta_2)^T, x^{(t)} + (-\delta_1, \delta_2)^T, x^{(t)} + (-\delta_1, -\delta_2)^T\}$. Using the elements of $S^t$, we define the following matrices:

$$W(t) = \begin{pmatrix} 1 & x_1^T \\ 1 & x_2^T \\ \vdots \\ 1 & x_{n}^T \end{pmatrix}, \quad y(t) = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$ \hspace{1cm} (8)

where $y_i$ is the observation corresponding to the point $x_i$. Then, we estimate the gradient by

$$\tilde{g}^t = \left( (W(t))^T W(t) \right)^{-1} (W(t))^T y(t).$$  \hspace{1cm} (9)

B. Multi-Armed Bandits (MAB) Algorithms

This type of algorithms consider a finite space of actions. We will use the same discrete set $\mathcal{A}$ considered in our proposal. Algorithm 5 details the general framework of a MAB algorithm. The differences among the algorithms are the initialization parameter $\bar{n}$ and the policy to select the next action. Note that each iteration of the for loop in Algorithm 5 corresponds to one stage.

**$\varepsilon$-greedy policy** [34] selects with probability $1 - \varepsilon$ the action with higher average performance and, with probability $\varepsilon$, a randomly chosen action. The probability $\varepsilon$ is a fixed value and the initialization parameter $\bar{n} = 1$.

Similarly, **$\varepsilon$-greedy logarithmic descent** policy uses $\varepsilon = 1/k$ and **$\varepsilon$-greedy logarithmic descent** policy uses $\varepsilon = 1/\log(k)$ where $k$ is the number of the current stage.

Fig. 5. Simulated scenario considering 4 pico eNBs per sector (dots) and highlighting the evaluation sectors (dark grey).

VII. Numerical Evaluation

In this section we compare our proposal against the benchmark algorithms both under stationary and variable network conditions. Moreover, we address the adjustment of the configurable parameter $\tau$.

A. Description of the Simulated Scenario

The simulated scenario is based on the 3GPP guidelines for LTE performance evaluation [27]. The wireless channel is modeled with deterministic pathloss attenuation and random shadow fading. The simulation layout is shown in Fig. 5, comprising 57 sectoral macro eNBs of 120 degrees (19 macro eNBs), and the
TABLE IV

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network layout</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>System bandwidth</td>
</tr>
<tr>
<td>LTE frame duration</td>
</tr>
<tr>
<td>Transmit power</td>
</tr>
<tr>
<td>Macro sector antenna pattern</td>
</tr>
<tr>
<td>Pico antenna pattern</td>
</tr>
<tr>
<td>Antenna gain</td>
</tr>
<tr>
<td>Macro to UE path loss</td>
</tr>
<tr>
<td>Pico to UE path loss</td>
</tr>
<tr>
<td>Shadow fading</td>
</tr>
<tr>
<td>Thermal noise</td>
</tr>
<tr>
<td>Scheduling algorithm</td>
</tr>
<tr>
<td>Traffic model</td>
</tr>
<tr>
<td>File size</td>
</tr>
<tr>
<td>$\lambda$ [UBs/s] (Offered traffic load [Mbps])</td>
</tr>
<tr>
<td>Minimum distances</td>
</tr>
</tbody>
</table>

simulation parameters are shown in Table IV. Several pico eNBs overlaps each eNB sector (shown as dots in Fig. 5), and pico eNBs can be (de)activated over time due to, for example, energy saving mechanisms.

The simulations are executed for the central 21 sectorial macro eNBs (dark grey zone in Fig. 5), while the rest of macro eNBs emulate the interference of a larger network. The total interference at each UE receiver is the aggregation of all interfering eNBs in the sector (macro and picos) plus the interference from the four nearest sectorial macro eNBs.

In the simulation, UEs arrive following a Poisson process of rate $\lambda$ arrivals per second on each sector. When considering dynamical network conditions, $\lambda$ varies during simulation time. According to the 3GPP FTP traffic model [27], each incoming UE downloads one file, and then leaves the network. In addition, each UE is dropped uniformly over the macro eNB coverage area with probability $\frac{1}{3}$, or over a pico eNB coverage area with probability $\frac{1}{5}$.

### B. Evaluation Procedure

In order to evaluate the algorithms, we follow the next procedure:

- First, the associated performance $F(x)$ is evaluated for each configuration $x \in \mathcal{A}$. We consider that the set $\mathcal{A}$ has a total of 285 components, as a consequence of the combination of 15 and 19 values for $\gamma$ and $\phi$, respectively. Evaluation is performed by Monte Carlo, running 40 trials for each configuration.

- A Normal distribution is assumed to characterize $F(x)$ at each point $x$. Its mean $\mu_F(x)$ and variance $\sigma^2_F(x)$ are obtained from their MSE estimators from the samples obtained above.

- For configurations $x$ outside the set $\mathcal{A}$, mean and variance are approximated by the MSE polynomial fits of degree 4 to the results obtained in the previous step. Henceforth, let $P\mu$ and $P\sigma^2$ denote these polynomial fits.

- The optimal configuration is determined as point $x^*$ (see\(^2\)) maximizing $P\mu$.

Then, to evaluate performance for a given algorithm, the states trajectory $x_1, x_2, \ldots, x_k, \ldots$ is determined by: (i) sampling the Gaussian process $F(x_k) \sim N(\mu(x_k), \sigma^2(x_k))$ to obtain sample $y_k$, (ii) determining $x_{k+1}$ from $x_k$ and $y_k$ following the corresponding algorithm. For each stage $k$, the regret is updated by adding $P(\mu(x^*) - \mu(x_k))$. The total regret is computed by performing $5$ independent trajectory runs each comprising 5000 stages.

### C. Stationary Network Conditions

In this setup we assume that network conditions (traffic intensity and number of active pico eNBs) do not change during evaluation, but UE generation is still a stochastic process as described above. Besides, initial state $x_1$ is set $(1.5, 1.5)$.

To optimize performance, SGA algorithms have been configured with a decreasing step-size following the function $c / \sqrt{t}$, where $c$ is a constant and $t$ is the current stage. The value of $c$ has been set to minimize the regret in Scenario 2, and has been obtained by exhaustive simulation. Moreover, for all algorithms $\delta = 1$ except RSM, which uses $\delta = 0.25$.

Regarding the configurable parameters of the MAB algorithms, $c$-greedy is configured with $c = 0.15$, softmax with $\xi = 0.005$, softmax logarithmic descent with $\xi_0 = 1$ and Thompson Sampling with uniform prior for $\sigma$ [30]. Our proposal is evaluated considering a fixed $m = 4$ and the best value of $\tau$ obtained by exhaustive simulation for each scenario.

At this point, we want to highlight the importance of the regret in an online setting. The regret accumulates, at each stage, the distance from the expected performance of the current action to the expected performance of the best action (1). That is, the farther from the best performance, the higher the regret is. Besides, this metric gives us information about something similar to the convergence of the algorithm. Note that we cannot talk about convergence rigorously in a non-stationary online setting since the optimal action changes over time. For this reason, our objective is to be as close as possible to the optimal action at every stage. It can be seen in the regret: when the slope of the regret curve is close to zero, the algorithm is very close to the optimum. However, the higher the regret slope, the larger the distance between the actions.

\(^2\) $x^*$ is determined to allow regret computation within the simulator, but it is not required in a real operating network.
selected by an algorithm and the optimal action in terms of performance.

Fig. 6 shows the numerical results for Scenarios 1, 2 and 3 in term of regret and Fig. 7 shows a trace of the 5th percentile throughput in Scenario 1. Note that the performance of each algorithms with respect to others changes depending on evaluated scenario. That is because the system response and its variance changes with the scenario and therefore the optimal values of the configuration parameters of each algorithm such as the step-size, \( \lambda \), \( \varepsilon \), etc. See for example that RSM algorithm is not the best option in Scenario 1 but its performance increases in Scenario 3 with respect to the others. Something similar occurs with the Softmax policy. Finding the optimal parameter configuration for each algorithm in each scenario is a very complicated task. Even more when we consider a real operating network, where the scenario (system response) is continuously chang-
Fig. 7. Numerical evaluation in a network with fixed conditions. Performance in terms of 5th percentile throughput for Scenario 1. (a) Stochastic gradient ascent algorithms. (b) Multi-armed bandit algorithms.

Fig. 8. Numerical evaluation considering the throughput as the objective performance metric. Performance in terms of regret for Scenario 1. (a) Stochastic gradient ascent algorithms. (b) Multi-armed bandit algorithms.

Note that, for MAB algorithms, the simpler policies ($\varepsilon$-greedy and softmax) obtain better results than sophisticated ones (UCB normal and Thompson sampling normal) in our setting. In the case of the UCB Normal policy, it has to explore by definition each action at least $\lceil 8\log(k) \rceil$ times at stage $k$ and, given the total number of actions in our setting ($|A| = 285$), it behaves like a random policy. On the other hand, although the Thompson sampling policy finally finds the region of the optimal configuration (the slope of its regret tends to zero) it takes more stages than other algorithms. However, $\varepsilon$-greedy descent policy which is one of the simplest policies, obtains a low regret in all scenarios.

It is worth highlighting that, although our proposal considers only a finite set $A$ of configurations, which in general does not contain $x^*$, it obtains lower regret than SGA algorithms which operates over the whole set $P$ and could, in principle, converge to $x^*$. In other words, the configurations selected by our proposal are, in average, closer to the optimum than the configurations selected by SGA algorithms.

As we claim in Section IV-A, our proposal can operate regardless of the selected performance metric $F(x)$. To illustrate that, we run our algorithm considering the average throughput instead of the 5th percentile throughput as our objective performance metric. We consider the same configuration parameters described in this section. Figs. 8 and 9 show the performance in terms of regret and throughput for Scenario 1, respectively. Note that, although MAB algorithms show a similar performance as in Fig. 6, SGA algorithms perform significantly worse. A parameter tuning of SGA algorithms may increase their...
Fig. 9. Numerical evaluation considering the throughput as the objective performance metric. Performance in terms of throughput for Scenario 1. (a) Stochastic gradient ascent algorithms. (b) Multi-armed bandit algorithms.

Fig. 10. Numerical evaluation in a network with changing conditions. We switch from Scenario 1 to Scenario 2 at stage 2500. (a) Stochastic gradient ascent algorithms. (b) Multi-armed bandit algorithms.

performance. However, our proposal obtains always the lowest regret regardless the scenario and the selected performance metric without tuning any parameter.

D. Dynamic Network Conditions

In this case, we assume changes in the network conditions in runtime: We switch from Scenario 1 to Scenario 2 (described in previous section) at stage 2500. This change would be produced, for example, by an energy saving mechanism which activates or deactivates pico eNBs depending on the network load. A different scenario implies, in general, a different system response, and therefore a change on the region where the most efficient configurations are located. Consequently, after stage 2500 we consider the new optimal configuration $x^*$ in the regret computation. Regarding parameter configuration of the algorithms, we consider the same of previous section and $\nu = 5$ for the NCD submodule.

Fig. 10 shows that our proposal also obtains lower regret than other algorithms in a dynamic setting. Benchmark algorithms are not aware of the change in network conditions, which affects their performance negatively. For example, SGA algorithms diminish their step-size at each iteration. This assures convergence under stationary conditions, but if these conditions undergo a significant change, the step-size should be restarted. A usual solution to this issue is to use a fixed step-size which can help to track the changes in system response, in exchange for worse performance under stationary conditions.

MAB algorithms are also designed for stationary scenarios, in which they aim at finding the best performing action based on the observed history, as this history becomes sufficiently long. However, when switching from Scenario 1 to Scenario 2, past history becomes not only irrelevant but misleading for MAB algorithms, since they keep selecting the arms that were identified as optimal for Scenario 1 but perform poorly in Scenario 2 [see Fig. 10(b)]. One solution for MAB algorithms is to be restarted when the network conditions changes, forgetting all the information collected until this point. It implies that the MAB initialization phase will select $\bar{n}$ times all actions in $A$. 
In contrast, when our proposal detects a change in system response using the NCD module, the search period restarts in the AAS module, which only selects actions nearby the current one. In order to evaluate the system responsiveness, we measured empirically the number of stages that NCD needs in average to detect the variation on the network conditions. For that purpose, we evaluated multiple simulation runs these three scenarios obtaining always similar results, a quick adaption which takes in average 26 stages. Moreover, the number of false positives (detect a erroneous change in the scenario) was null in all the simulations.

E. Comparison to a Model-Driven Approach

The regret of an algorithm essentially gives us the performance gap of the algorithm with respect to the best possible configuration. However, in order to illustrate the benefits of our model-free approach, we compare our proposal to an existing model-based scheme: PF-ABS [15]. Table V shows the performance values attained by each algorithm in terms of 5th and 50th percentile throughput in three different scenarios characterized by the number of pico eNBs per sector (10, 14 and 18). DDeC outperforms PF-ABS in terms of 5th percentile throughput in all scenarios, while PF-ABS achieves better results in terms of 50th percentile throughput only for low density scenarios. These results are consistent with the fact that DDeC was configured to optimize the 5th percentile throughput, but its optimization objective can be changed to other metric (see Section VII-C). In contrast, PF-ABS is not able to change its optimization objective.

In addition, PF-ABS assumes PF as a scheduling algorithm and, using the PF metric, configures the ABS ratio of each eNB. As a result, PF-ABS uses local configurations. This allows a distributed implementation but, in general, it tends to perform worse than global configurations (see Section V-D). In order to compare both schemes fairly, we show the PF-ABS results using the optimal CRE bias for each scenario.

We would like to emphasize that our scheme does not necessarily replace model-based approaches. Instead, both approaches can complement each other. For example, in a plug-and-play scenario, a model-driven algorithm can be used to find an efficient configuration given the current network conditions. Then, our data-driven mechanism can use this configuration as an initial point to perform a fine parameter tuning and track the optimal configuration over time.

F. Evaluation of \( \tau \) Parameter

As discussed in Section V-A, \( \tau \) balances the accuracy and the adaption velocity of our algorithm. For example, a large value of \( \tau \) implies a more accurate selection of the best local action but taking more time in each iteration, as a tradeoff.

The optimal value of this parameter (\( \tau^{*} \)) is the one that minimizes the total regret of our proposal and depends on the system response (network conditions). By simulation, we found that the values of \( \tau^{*} \) are 34, 31 and 29 for scenarios 1, 2 and 3, respectively. However, since the system response is a priori unknown and changes continuously over time, we are no able to find \( \tau^{*} \) at every step, specially in a real operating network. For this reason, we evaluate the average relative loss\(^3\) of using \( \tau \) (the average of the optimal values for these tree scenarios) as a fixed value instead of \( \tau^{*} \) of each scenario.

Our numerical results show that, although the system response and the optimal configuration of the network changes with network conditions (e.g., traffic load, pico eNB topology), we can configure our algorithm with \( \tau \) as a fixed value not implying noticeable losses (0.0007\% per stage).

VIII. CONCLUSION

In this paper, we have presented a Data-Driven eICIC Conﬁgurator (DDeC) for interference coordination in HetNets environments based on a big data-driven framework for network optimization. It ﬁnds efﬁcient conﬁgurations of eICIC parameters (ABS ratio and CRE bias) operating without any previous knowledge of the network (e.g., traffic load, topology, propagation conditions) only using data retrieved from the network. This data-driven approach allows us to optimize metrics of practical interest that are analytically intractable, such as 5th percentile throughput. Our numerical results show that our proposal achieves a signiﬁcant improvement with respect to the state of the art of SGA and MAB algorithms in networks with both stationary and variable conditions.

REFERENCES


Jose A. Ayala-Romero received the B.Sc. degree in telecommunication engineering and the M.Sc. degree in telecommunication engineering from the Technical University of Cartagena, Cartagena, Spain, in 2012 and 2014, respectively, where he is currently working toward the Ph.D. degree with the Department of Information and Communications Technologies. His research interests include cellular networks and learning algorithms.

Juan J. Alcaraz received the M.Sc. degree in telecommunications engineering from the Polytechnic University of Valencia, Valencia, Spain, in 1999, and the Ph.D. degree in telecommunications engineering from the Technical University of Cartagena (UPCT), Cartagena, Spain, in 2007. Between 1999 and 2004, he was employed by several technology companies. In 2004, he joined UPCT, where he is currently an Associate Professor with the Department of Information and Communication Technologies. He was a Fulbright Visiting Scholar with the Electrical Engineering Department, University of California, Los Angeles, in 2013, and a Visiting Professor with the Department of Information Engineering, University of Padova, in 2017. His current research focuses on learning algorithms for wireless network management.

Javier Vales-Alonso received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Vigo, Vigo, Spain, and the Technical University of Cartagena, Cartagena, Spain, in 2000 and 2005, respectively, and the M.Sc. degree in mathematics from the National University of Distance Education, Madrid, Spain, in 2015. His research interests include modeling and optimization.
Energy Saving and Interference Coordination in HetNets Using Dynamic Programming and CEC

JOSE A. AYALA-ROMERO*, JUAN J. ALCARAZ*, AND JAVIER VALES-ALONSO*
Department of Information and Communications Technologies, Technical University of Cartagena, 30222 Cartagena, Spain
Corresponding author: Jose A. Ayala-Romerom (josea.ayala@upct.es)

This work was supported by Project Grant AEI/FEDER TEC2016-76465-C2-1-R (AIM). The work of Jose A. Ayala-Romero was supported by personal Grant FPU14/03701.

ABSTRACT Energy efficiency in cellular networks has gained great relevance due to the increasing power supply demands in new generation heterogeneous network (HetNet). On the other hand, interference coordination is a key aspect in HetNets resource management which directly affects the performance and energy consumption. Although these two aspect are intertwined, they have been studied separately so far. In this paper, we address the joint problem of energy saving and interference coordination in HetNets. We formulate the problem as a finite horizon Markov decision process (MDP) leveraging two facts: the user traffic demands usually follow periodic patterns, and the knowledge and prediction of network load is crucial in order to select efficient network configurations. Quality of Service (QoS) in the network is defined as the ratio of users meeting a requirement specified by the operator, and allows us to account for a minimum QoS requirement by including a constraint in the formulation of the MDP. To address this MDP, we propose an approximate dynamic programming (ADP) algorithm which selects energy efficient farsighted configurations with QoS guarantees achieving near optimal performance. This ADP algorithm is built upon: 1) the certainty equivalent control principle, which simplifies the complexity of the MDP, and 2) machine learning techniques: a neural network and a polynomial regressor which allow us to predict the QoS and the consumption of the network in advance. We evaluate our proposal in a LTE-A network simulator following the 3GPP guidelines, and the results obtained show that a joint control of the energy saving and interference coordination mechanisms results in a notably performance improvement compared to a disjoint control, in terms of both energy savings and QoS guarantees. Moreover, our proposal has the advantage of being adaptable to the operator QoS requirements.

INDEX TERMS Dynamic programming, energy saving, green networking, heterogeneous networks, interference coordination.

I. INTRODUCTION

One of the most prominent features of 5G networks is expected to be a dense deployment of small cells to increase the spatial spectrum reuse [1]. However, small cell densification entails two problems: a higher energy consumption in the access network infrastructure, and an increased intercell interference [2]. These two issues have been addressed separately before, but they should not be decoupled because of two reasons: first, the energy saving mechanisms can affect the interference profile (particularly when these mechanisms switch base stations on or off). And second, because the interference coordination mechanism affects the traffic load and the available spectral resources at each station, which are aspects that the energy saving policies should consider and exploit.

This paper addresses energy saving at the base stations and interference coordination as a joint problem, using a stochastic control approach. We show that, addressing both problems simultaneously, a considerable improvement in energy saving is achieved.

Micallef et al. [2] explain that over 80% of the energy consumed in a mobile network is due to the base stations. There are, however, two aspects enabling energy saving strategies: the overlapping coverage areas of many cells and the variable traffic demands. By exploiting these aspects, some underutilized cells can be switched off without perceptible degradation on the user QoS. This strategy outperforms other approaches based on transmission power optimization, which incurs in load-independent energy consumption [3].
We consider a heterogeneous network (HetNet) composed of pico eNBs overlapping the macro eNB coverage area in each sector. The pico eNBs can be in active or sleeping mode (on or off), according to the energy saving mechanism, while the macro eNB is always active. The energy consumption model in this work is load dependent and considers an activation cost when a sleeping pico eNB is switched on. We apply the enhanced Inter Cell Interference Coordination (eICIC) mechanism defined by the 3GPP for LTE-A Networks [4]. The eICIC parameters are Cell Range Expansion (CRE) bias and Almost Blank Subframe (ABS) ratio.

We formulate the problem as a finite horizon Markov Decision Process (MDP) encompassing the daily pattern of traffic demands [5], [6]. This allows us to capture the variable traffic conditions during each day. However, the use of this formulation and inserting the eICIC parameters in the control space increases notably the complexity of the problem, compared to action spaces restricted to energy decisions [7], [8].

To address this complex problem, we propose an Approximate Dynamic Programming (ADP) algorithm that combines stochastic control and machine learning techniques. It comprises an offline phase where the cost-to-go function can be efficiently estimated from stored network data, and an online phase, where the algorithm selects, at each decision stage, efficient farsighted controls (considering their impact in future decision stages) with QoS guarantees.

The remainder of the paper is organized as follows. In section II the related work and contribution are discussed. In section III we describe the power consumption model and the interference coordination mechanism. The system model and the formulation of the problem are given in Section IV. In Section V we describe the proposed mechanism. The simulation framework and the numerical results are given in Section VI. Finally, the conclusions are summarized in Section VII.

II. RELATED WORK AND CONTRIBUTION

A. RELATED WORK

The optimization of energy efficiency decisions in HetNet has been previously addressed as a stochastic control problem. The most usual approach is to formulate the problem as a Markov Decision Process (MDP) including restrictions that account for the QoS objectives [7], [8], [21] as in our proposal. The inherent computational complexity of the MDPs implies the use of ADP approaches like Reinforcement Learning (RL) [21]. Other works [24] combine game theoretic techniques with RL algorithms.

Similarly, the stochastic control approach has also been used to address interference coordination in heterogeneous networks [25]–[27]. However, as far as we know, our work is the first addressing the joint control of energy saving and interference coordination in HetNets as a stochastic control problem. The improvement associated to the inclusion of interference coordination control can be seen in our numerical results in Section VI-D. Previously, Virdis et al. [18] addressed the energy saving problem in HetNets exploiting the ABS configuration, which strongly indicates that it has a significant impact in the power consumption, as our hypothesis. However, user association and eNB on-off switching are not considered in this work.

A common simplification (e.g. [7], [8], [21]) is to model the traffic intensity as a homogeneous Markov process. Nevertheless, real network traces exhibit a daily pattern [5], [6], [28]. Thus, we aim at exploiting this fact by formulating a finite horizon MDP with one-day horizon. This formulation allows us, in contrast to previous works, to apply the certainty equivalent control (CEC) which is an approximate dynamic programming technique to approximate the cost-to-go function with a deterministic model [29].

In order to reduce the problem dimensionality some works make a compact representation of the state space using a function approximation [21]. In our case, the control space is specially large because it comprises not only energy saving but also interference coordination decisions. For this reason, we propose to transform the state and control spaces into reduced dimensional ones achieving a notable reduction of the computational cost of both offline and online phases.

Other works [9]–[16], [18], [30] address the eNB on-off switching problem as an optimization problem. Since the computational complexity of these problems is NP-Hard, they are addressed using iterative algorithms or decomposing the problem in relaxed ones aimed at finding operative (but suboptimal) solutions [31]. An additional issue is that these solutions have to be recomputed whenever the state of the network changes (e.g. traffic intensity). In contrast, our solution operates for any network state. Some of the aforementioned works also consider the user association [10]–[13], but none of them take into account the interference management, which is an essential issue in HetNets.

The eNB switching problem has been also addressed using Stochastic Geometry [17], [19], [20], [32], [33]. However, this approach implies some simplifications in the network model (e.g. path loss as a channel model, eNBs deployed following a Poisson Point Process). In contrast, our proposal is capable of using real data from the network in both offline and online phases and does not require any simplification, neither does assumptions in the network model.

Regarding the power consumption model, some works [5], [28], [34] consider two values of power consumption (cell on or off). Other works [7], [21], [32] consider a more detailed model where the power consumption of a cell depends on its traffic load. In our work, we consider the load dependent consumption model proposed by the 3GPP [35] with additional consumption spikes associated to the activation of sleeping eNBs [6].

B. CONTRIBUTION

The main contributions of this paper are:

- To consider the joint control of energy saving and interference coordination mechanisms in HetNets leading to
TABLE 1. Comparison of energy saving related works.

<table>
<thead>
<tr>
<th></th>
<th>[5], [9]</th>
<th>[10]</th>
<th>[11]–[16]</th>
<th>[17]</th>
<th>[18]</th>
<th>[19]</th>
<th>[20]</th>
<th>[21]</th>
<th>[7]</th>
<th>[22], [23]</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>eNB on-off Switching</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>User Association</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Power Control</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Interference Management</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Approach</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>SG</td>
<td>O</td>
<td>SG</td>
<td>O</td>
<td>RL</td>
<td>O</td>
<td>DP</td>
<td></td>
</tr>
</tbody>
</table>


remarkable performance improvement compared to a disjoint control.

• To formulate the problem as a constrained finite horizon MDP leveraging two facts: the traffic demands usually follow periodic patterns, and the knowledge and prediction of network load is crucial in order to select efficient network configurations.

• To propose an ADP algorithm achieving near optimal performance in terms of energy consumption and QoS of the users, reducing energy consumption up to 24% with respect to a fixed default configuration.

• To build the ADP algorithm using a novel combination of stochastic control and machine learning techniques. In particular, we use Certainty Equivalent Control (CEC) in combination with Neural Networks and polynomial regression.

III. INTERFERENCE MANAGEMENT AND POWER CONSUMPTION MODEL

A. INTERFERENCE MANAGEMENT IN LTE-A: eICIC

The eICIC is an interference management mechanism for heterogeneous networks defined by the 3GPP from Release 10 (LTE-A) [4]. It schedules the radio resources for macro and pico eNBs in different time periods (called subframes) to avoid inter-cell interference. The eICIC comprises the following features: Cell Range Expansion (CRE) and Almost Blank Subframe (ABS).

Pico eNBs are low transmission power cells aimed at increasing the spatial reuse of radio resources. However, LTE cell association is based on the received signal reference power (RSRP), that is, each UE associates with the highest reference signal received from each cell. Therefore, UEs located close to a pico could end up associating to the macro eNB, precluding spatial reuse. For this reason, CRE allows pico eNBs to extend their footprint by adding a bias to their RSRP.

An associated issue is that the UEs located at pico eNB extended region (CRE region) will, in general, have a low channel quality due to the high interference received from macro eNB. In order to improve that quality eICIC allows to mute all data symbol in certain subframes (ABS subframes). Then, the UEs located at CRE region can be scheduled in ABS subframes obtaining a higher signal-to-interference-plus-noise ratio (SINR) due to the absence of interfering signal from the macro eNB. The ABS pattern has a periodicity of 8 subframes. We define the ABS ratio as the portion of muted subframes over conventional ones.

B. ENB POWER CONSUMPTION MODEL

In this section, we describe the eNB consumption model used in this work. It is based on the 3GPP guidelines [35] and also includes the consumption associated to the activation of a sleeping eNB [6].

The eNB transceivers (TRXs) are composed of an AC-DC unit for main supply, a power amplifier, a cooling system, a baseband interface (a receiver for downlink and a transmitter for uplink), and an antenna interface. The power consumption of some of these components depends on the load of the eNB. Therefore, a common assumption is to approximate the relation between RF output power and power consumption of TRXs using a linear model [35]. Then, the power consumption model of a pico eNB $j$ is given by:

$$C_p^j = e^j \cdot N_{TRX} \cdot (P_0 + R^j \cdot P_{max}) + (1 - e^j) \cdot N_{TRX} \cdot P_{sleep} + \Delta$$

$$\Delta = \begin{cases} \beta \cdot P_0 & \text{when eNB } j \text{ is switched on from sleeping mode} \\ 0 & \text{otherwise} \end{cases}$$

where:

• $e^j = 1$ when the pico eNB $j$ is active and $e^j = 0$ otherwise.

• $N_{TRX}$ is the number of TRXs.

• $P_{max}$ is the maximum RF output.

• $P_0$ represents the power consumption at zero RF output power.

• $R^j \in [0, 1]$ is the load factor of the pico eNB $j$ and depends on the ABS ratio, the CRE bias, the traffic intensity and the location of UEs.

• $P_{sleep}$ is the power consumption of TRX components in sleep mode.

• $\beta$ is the portion of $P_0$ needed to switch on the pico eNB TRXs.

Note that $\Delta$ captures the consumption associated to switching on a sleeping pico eNB. The power consumption of the macro eNB $i$ is given by

$$C_m^i = N_{TRX} \cdot (P_0^m + R^i \cdot P_{max}^m) \cdot (1 - \gamma) + N_{TRX} \cdot P_{sleep}^m \cdot \gamma$$

(3)
where $\gamma$ denotes the ABS ratio, $P_{\text{max}}^\alpha$ is the maximum power output of the macro eNB and $P^\phi_0$ is the power consumption at zero RF output power of the macro eNB. Note that the ABS ratio, the CRE bias and the activation controls of the eNBs can affect the load factor $\mathcal{P}_f$ of an eNB $j$, making the global power consumption in the network an unknown non-convex function. This motivates the inclusion of these parameters in the control of energy consumption, as explained in next Section.

IV. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

We consider a sector of a LTE-A access network comprising a macro eNB and $P$ pico eNBs. Let $\mathcal{P} = \{1, 2, \ldots, P\}$ be the set of pico eNBs overlapping the macro sector.

The ABS ratio and the CRE bias are denoted by $\gamma \in \Gamma$ and $\phi \in \Phi$, where $\Gamma$ and $\Phi$ are the finite sets of all possible configurations of these parameters. Time is divided into discrete time stages $k \in \{0, 1, \ldots\}$. At each stage, the decision maker must select the values for $\gamma$, $\phi$ and the activation state of the pico eNBs in $\mathcal{P}$. This sequential decision procedure is modeled by a Markov Decision Processes characterized by the following elements:

1) States: Let $e^d_k$ be the state of the pico eNB $j \in \mathcal{P}$ at stage $k$, where $e^d_k = 1$ when the pico eNB $j$ is switched on and $e^d_k = 0$ otherwise. The vector $p_k = (e^d_1, \ldots, e^d_P) \in \mathcal{E}$ ($\mathcal{E} = \{0, 1\}^P$) denotes the activation state of all picos in $\mathcal{P}$. Let $\lambda_k \in \Lambda$ refer to the mean number of UEs in the sector (traffic load) at stage $k$, where $\Lambda = [0, \lambda_{\text{max}}]$. The state of the system at stage $k$ is given by $x_k = (\lambda_k, p_{k-1}) \in \mathcal{X}$, where $\mathcal{X} = \Lambda \times \mathcal{E}$ is the state space. Thus, the state of the system is defined by the traffic load at current stage and the activation state of the pico eNBs in $\mathcal{P}$ established in the previous stage.

2) Controls: At the beginning of each stage, the decision maker observes $x_k$ and makes a decision regarding which stations are switched on and off, as well as the values of $\gamma$ and $\phi$ to be used during stage $k$. These decisions define the control $u_k = (p_k, \gamma_k, \phi_k) \in \mathcal{U}$, where $\mathcal{U} = \mathcal{E} \times \Gamma \times \Phi$ is the control space.

3) State transitions: The state of the system at next stage is given by $x_{k+1} = f_k(x_k, u_k)$, where $f_k$ is a function defining the stochastic state transition. In our case, the state transition can be written as follows: $x_{k+1} = (\lambda_{k+1}, p_k)$, where the random variable $\lambda_{k+1}$ entails the stochastic nature of state transition. Fig. 1 depicts the state transition.

4) Payoff functions: We define two global payoff functions: $C : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^+$ providing the aggregated power consumption in the network and $Q : \mathcal{X} \times \mathcal{U} \to [0, 1]$ which gives us the ratio of UEs in the network satisfying the QoS requirement specified by operator. In particular, the QoS requirement is the UE throughput being greater than a given (specified) threshold. Therefore, the QoS function indicates the ratio of UEs whose throughput is greater than the specified threshold. Note that $C$ and $Q$ are random since they depend on the random UE locations and traffic demands.

B. FUTURE-HORIZON DP PROBLEM FORMULATION

It has been widely reported that data traffic in a cellular network presents a periodic pattern [5], [6], [28], according to which the average traffic load can be described by a periodic function with a one-day period length. We leverage this fact by formulating the problem as a finite-horizon MDP where the length of the horizon is one day. We divide the day into $N$ discrete time stages $k = 0, 1, \ldots, N - 1$. An action $u_k$ is selected for each state $x_k$ for each $k = 1, \ldots, N - 1$. Then, at the end of the $N$ stages, we use the boundary control $u_N = (p_N, \gamma_N, \phi_N)$ for resetting the process to an initial state $x_0 = (\lambda_0, p_0)$.

The mapping $\pi_k : \mathcal{X} \to \mathcal{U}$, determines the action $u_k$ at stage $k$ given the current state $x_k$. A policy for the finite-horizon MDP is given by $\pi = \{\pi_0, \ldots, \pi_{N-1}\}$. The objective is to find the optimal policy $\pi^*$ minimizing the expected accumulated consumption over the $N$ stages and satisfying the QoS constraint at every stage:

$$
\pi^* = \arg\min_{\{\pi_0, \ldots, \pi_{N-1}\}} \mathbb{E} \left[ C(x_N, u_N) + \sum_{k=0}^{N-1} C(\lambda_k, p_{k-1}, \pi_k(x_k)) \right]
$$

s.t. $\mathbb{E} \{Q(\lambda_k, \pi_k(x_k))\} > Q_{\text{min}}$ for $k = 0, \ldots, N - 1$

(4)

where $Q_{\text{min}}$ defines the QoS requirement and $C(x_N, u_N)$ is the consumption associated to resetting the system for the next day using the boundary control $u_N$.

Let $J_k(x_k)$ be the cost-to-go which indicates the minimal consumption of the system from stage $k$ to the final stage $N$, provided that, at stage $k$, the system is in state $x_k$ and considering that the QoS requirement is met at every stage.

$$
J_N(x_N) = C(x_N, u_N)
$$

$$
J_k(x_k) = \min_{u_k \in \mathcal{U}} \mathbb{E} \left[ C(x_k, u_k) + J_{k+1}(f_k(x_k, u_k)) \right]
$$

s.t. $\mathbb{E} \{Q(x_k, u_k)\} > Q_{\text{min}}$ for $k = 0, \ldots, N - 1$

(5)

The computational overhead associated to the computation of the cost-to-go may be prohibitively large. The reason is twofold: first, the dimensions of $\mathcal{X}$ and $\mathcal{U}$ grow exponentially with the size of $\mathcal{P}$, which also limits the scalability of the problem. Second, the stochastic nature of state transition makes it necessary to compute, for each stage $k$, the probability of all possible traffic intensities $\lambda_{k+1}$ at next stage. Specifically, considering $\Lambda$ as a finite set of traffic intensities, the cost-to-go function consists of $|\Lambda| 2^P$ values. To obtain each of these values, we need to compute the expected values of the consumption and the QoS for each of the $2^P \cdot |\Gamma| \cdot |\Phi|$
controls in $\mathcal{U}$. Note that $\Lambda$ has been defined as a continuous set in Sec. IV-A. In this case, the cost-to-go is be a continuous function for each value of $k$. In the next section we present our proposal aimed at avoiding these problems.

V. PROPOSED CONTROL SCHEME

Our proposal to address problem (4) in a computationally feasible manner, comprises the combination of the following techniques: first, we make use of the Certainty Equivalent Control (CEC) principle to reduce the complexity associated to the stochastic nature of the state transitions. Second, we transform the state and control spaces into a reduced dimensional ones. Third, we use a machine learning approach to infer instant power consumption and to evaluate QoS constraint satisfaction.

In the offline phase we obtain an estimator for the consumption, $\hat{\mathcal{C}}$, and an estimator of the QoS, $\hat{\mathcal{Q}}$. These estimators should be trained offline with a data set comprising user performance samples and network energy consumption for diverse states and configurations of the system. This data set can be readily available to networks operators from the real network, testbeds and/or simulation. Using the estimators $\hat{\mathcal{C}}$ and $\hat{\mathcal{Q}}$ and the CEC principle (Sec. V-A), the estimation of the Cost-to-go function $\hat{J}$ is computed (Sec. V-D).

Fig. 2 shows the operation of the online phase where the ADP algorithm entity controls a cluster comprising $M$ sectors. The ADP algorithm receives the system state $x_k$ at each stage and, using $\hat{\mathcal{C}}$, $\hat{\mathcal{Q}}$ and $\hat{J}$, computes the dimensionally-reduced global control $u_k^g$ which is sent to the Control Space Augmentors (CSAs) (Sec. V-B). The $i$-th CSA (corresponding to the $i$-th sector) computes the full-dimensional control $u_k^i$ from $u_k^g$ using the specific information about the network topology of the $i$-th sector. Each sector $i$ operates during the stage $k$ using the control $u_k^i$ computed by its corresponding CSA. This semi-distributed approach reduces the dimensionality of state/control spaces, simplifying notably the computational complexity of the problem. For the sake of simplicity in the notation, we formulate the problem for only one sector. Note that, while the offline phase is performed once, the online phase is continuously executed in a stage by stage basis. The following subsections describe and discuss the elements of our proposal in detail.

A. CERTAINTY EQUIVALENT CONTROL (CEC)

The Certainty Equivalent Control (CEC) is an approximated control scheme that applies at each stage the control that would be optimal if the uncertain quantities were fixed to their expected values [29]. CEC differs from exact dynamic programming in the following aspects: the cost-to-go $J_{k+1}(x_{k+1})$ is replaced by its estimation $\hat{J}_{k+1}(x_{k+1})$, and the uncertain quantities are replaced with their expected values. Specifically, we replace the traffic intensity at each stage $k$, $\lambda_k$, with its expected value $\bar{\lambda}_k \in \Lambda$, where $\Lambda$ is a finite set whose cardinality is $|\Lambda| = N$ (it contains one expected value for each decision stage).

Therefore, considering $\bar{\lambda}_k$ at each stage $k$, the state transitions are now deterministic $x_{k+1} = \mathcal{P}(x_k, p_k)$, since $\lambda_{k+1}$ is known in advance and $p_k$ is provided by the control $u_k$.

The objective is to find the control that minimizes the sum of the expected power consumption at current stage and the cost-to-go estimation while satisfying the QoS requirement at each stage:

$$u_k = \arg\min_{u \in \mathcal{U}} E[C(x_k, u) + \hat{J}_{k+1}(\bar{\lambda}_{k+1}, u)]$$

s.t. $E[Q(x_k, u)] > Q_{\text{min}}$ for $k = 0, 1, \ldots, N - 1$. (6)

Note that this minimization is performed at each stage, while the network operates (this is referred to as online operation). Then, given the observed current state $x_k$, we need to predict the expectation of the energy consumption for all feasible controls: $E[C(x_k, u)] \forall u \in \mathcal{U}$, which is addressed later in this Section. Note at this point that the computational cost associated to solve (6) depends on $|\mathcal{U}|$. For this reason, we propose a state/control space transformation to make the problem computationally feasible.

B. STATE/CONTROL SPACE TRANSFORMATION

The possible values of the vector $p$ (which indicates the activation state of each pico in $\mathcal{P}$) increase exponentially with $P$: $|\mathcal{E}| = 2^P$. This section describes a strategy to reduce the relationship between $\mathcal{P}$ and state/control spaces from exponential to linear. We define $p' = \Pi(p)$ as the linear transformation of the parameter $p$ where $\Pi(x) = \sum_i x_i$. Thus, $p' \in \mathcal{E}'$ where $|\mathcal{E}'| = P + 1$. Let $u' = (p', \gamma, \phi) \in \mathcal{U}'$ denote a transformed control, where $\mathcal{U}' = \mathcal{E}' \times \Gamma \times \Phi$. This dimensionally-reduced control will use in next sections to define the estimation of the cost-to-go function and the online operation of our proposal. Although the transformation $\mathcal{U} \rightarrow \mathcal{U}'$ is straightforward, it must be undone in the last step of our algorithm (Fig. 2). CSA entities are in charge of reversing the transformation ($\mathcal{U}' \rightarrow \mathcal{U}$). In the reminder of this section we detail the procedure implemented by CSAs.
We define the cell adjacency value $d^i$ associated to a pico eNB $j \in \mathcal{P}$ as a weighted average of the distances to the remaining eNBs in the sector:

$$d^i = w \cdot d^i_{m} + (1 - w) \cdot d^i_{p}$$

(7)

where $d^i_{m}$ is the distance to the macro eNB, $d^i_{p}$ is the average distance to the rest of the pico eNBs in the sector and $w \in [0, 1]$ is a weighting factor. Let $\mathcal{D} = \{d^i\}_{i \in \mathcal{P}}$ be the set containing the cell adjacency values of all pico eNBs in $\mathcal{P}$. The higher the value of the cell adjacency, the higher the priority of a pico eNB to be switched off. Therefore, the transformation can be written as (for $p' \rightarrow p$), by computing the values of each element $e^i$ of $p$ as follows:

$$e^i = \begin{cases} 
1 & \text{if } i \in \mathcal{I} \text{ where } \mathcal{I} = M_{p'}(\mathcal{D}) \\
0 & \text{otherwise}
\end{cases}$$

(8)

where the operator $M_{p'}(\mathcal{D})$ returns the indexes of the $p'$ greatest values in $\mathcal{D}$, i.e., the indexes of the $p'$ pico eNBs with the greatest values of cell adjacency.

The idea behind this approach is the following. The cell adjacency value of a pico eNB indicates its closeness to the rest of the eNBs in the sector. Thus, our strategy consists on switching off firstly the eNBs with lower values of cell adjacency, that is, the eNBs with more eNBs in their surroundings because they can offload their traffic to near eNBs (macro or pico) incurring lower losses in the channel quality. We evaluate this strategy in Section VI.

C. MACHINE LEARNING FOR ENERGY CONSUMPTION AND QoS ESTIMATION

We define $\tilde{C} : \Lambda \times \mathcal{U'} \rightarrow \mathbb{R}$ as an estimator of the aggregated power consumption. The estimator $\tilde{C}$ is based on a polynomial regression model of the relation between the input variables (the system state and the selected control) and output variable (the observed power consumption). We evaluated polynomials of several degrees ranging from 1 to 9, using two data sets, one for training the model and other for testing its estimation accuracy. In these experiments, degree 6 polynomials obtained the minimum Root Mean Squared Error (RMSE) in the test data set.

Let $\tilde{Q} : \Lambda \times \mathcal{U'} \rightarrow [0, 1]$ be a classifier estimator indicating whether the current expected QoS is above the defined threshold $Q_{\min}$ ($\tilde{Q}(\lambda, u') = 1$) or not ($\tilde{Q}(\lambda, u') = 0$). The function $\tilde{Q}$ is implemented using a neural network (NN) whose architecture is shown in Figure 3. The number of nodes in the hidden layers are $H_1 = H_2 = 50$. A rectified linear unit (ReLU) is used as a nonlinear activation function in both hidden layers and we use a sigmoid activation function in the output layer. This neural network is trained using the gradient-based algorithm in [36] to minimize the cross-entropy cost function. The NN was trained during 400 epochs with a learning-rate equal to 0.001, using data previously gathered from the network. The data set used to train both models consists of 23000 samples of power consumption and QoS obtained by simulation.

D. COST-TO-GO ESTIMATION

In this section, we address the cost-to-go estimation presented in equations (6). First, we need to introduce some notation. Let $\tilde{x}_k = (\tilde{x}_k, \tilde{p}_{k-1}) \in \mathcal{X}$ be a transformed CEC state value, where $\mathcal{X} = \Lambda \times \mathcal{E}'$. We define $\Delta' : \mathcal{X} \times \mathcal{U'} \rightarrow \mathbb{R}$ as the function providing the power consumption associated to the activation of the pico eNBs. According to (2):

$$\Delta'(\tilde{x}_k, u') = \beta \cdot P_0 \cdot \max(0, p_{k} - p_{k-1})$$

(9)

The estimation of the cost-to-go at stage $k$ is defined as follows:

$$\tilde{J}_N(\tilde{x}_N) = \Delta'(\tilde{x}_N, u_N)$$

$$\tilde{J}_k(\tilde{x}_k) = \min_{u' \in U'} \tilde{C}(\tilde{x}_k, u') + \Delta'(\tilde{x}_k, u') + \tilde{J}_{k+1}(\tilde{x}_{k+1}, u_{k+1})$$

s.t. $\tilde{Q}(\tilde{x}_k, u') = 1$ for $k = 0, 1, \ldots, N - 1$.

(10)

where $\tilde{J}_k$ is the function defining the state transitions which are now deterministic and $\tilde{J}_N(\tilde{x}_N, u_N)$ correspond to the power consumption associated to the activation of the pico eNBs at the initial stage of the next day.

We summarize the offline phase as follows:

1. Retrieve the data set from the network whose $j$-th sample is $(x_j, u_j, C(x_j, u_j), \tilde{Q}(x_j, u_j))$.
2. Train the estimators $\tilde{C}$ and $\tilde{Q}$ according to the procedure detailed in Sec. V-C.
3. Compute $\tilde{J}$ using (10).

E. ONLINE OPERATION

In the online phase, we select the control $u'_k$ for each stage $k$ minimizing the power consumption afterwards and satisfying the QoS constraint, given the current state $x_k$ observed from the network, i.e.,

$$u'_k \in \arg \min_{u' \in U'} \tilde{C}(x_k, u') + \Delta'(x_k, u') + \tilde{J}_{k+1}(\tilde{x}_{k+1}, u')$$

s.t. $\tilde{Q}(\tilde{x}_k, u') = 1$ for $k = 0, 1, \ldots, N - 1$.

(11)

The online phase is summarized in Algorithm 1. It shows the operation of our proposal from a high-level point of view during one day. Steps 3, 4 and 5 in Alg. 1 are executed
TABLE 2. Notation table.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P )</td>
<td>Set of pico eNBs in the sector</td>
</tr>
<tr>
<td>( \gamma \in \Gamma, \phi \in \Phi )</td>
<td>ABS ratio, CRE bias</td>
</tr>
<tr>
<td>( p \in E )</td>
<td>Activation state of pico eNBs in the sector</td>
</tr>
<tr>
<td>( \lambda \in \Lambda )</td>
<td>Traffic intensity</td>
</tr>
<tr>
<td>( x = (\lambda, p) \in X )</td>
<td>Network state, network state space</td>
</tr>
<tr>
<td>( u = (p, \gamma, \phi) \in \mathcal{U} )</td>
<td>Control, control space</td>
</tr>
<tr>
<td>( C, Q )</td>
<td>Power consumption and QoS pay-off functions</td>
</tr>
<tr>
<td>( p' \in E' )</td>
<td>Activation state of pico eNBs in the sector</td>
</tr>
<tr>
<td>( u' = (p', \gamma, \phi) \in \mathcal{A} )</td>
<td>Transformed (dimensionally-reduced) control</td>
</tr>
<tr>
<td>( \bar{\lambda} \in \bar{\Lambda} )</td>
<td>Expected traffic intensity</td>
</tr>
<tr>
<td>( \bar{x} = (\bar{\lambda}, p') \in \bar{X} )</td>
<td>CEC state, CEC state space</td>
</tr>
<tr>
<td>( \bar{C}, \bar{Q} )</td>
<td>Power consumption and QoS estimators</td>
</tr>
<tr>
<td>( \Delta' )</td>
<td>Power consumption activation function</td>
</tr>
</tbody>
</table>

once per stage. Note that the computational complexity of solving (11) only depends on the size of \( \mathcal{U}' \), independently of the size of the network. This means that the computational complexity per stage in the online phase is \( O(1) \). That is, our proposal can be applied to an arbitrarily large network without increasing the computational complexity, only by replicating the instances of the ADP algorithm.

We propose to group the sectors in clusters characterized by a similar traffic profile (e.g., city center, business area). Thus, one ADP controller can be assigned to each cluster to cover the whole network. In other words, we need as many instances of the ADP algorithm as clusters of sectors in the network. Table 2 summarizes the notation in this paper.

Algorithm 1 Online Phase (ADP Algorithm Operation)

1: for \( k = 0, 1, \ldots, N - 1 \) do
2: Receive the network state \( x_k \)
3: Compute \( u'_k \) using (11)
4: Send \( u'_k \) to CSAs
5: end for

VI. NUMERICAL RESULTS
A. DESCRIPTION OF THE SIMULATION FRAMEWORK
The network layout is based on the 3GPP guidelines for evaluation of LTE networks [37]. It consists of 5 sectorial macro eNBs (120 degrees) and several pico eNBs overlapping each macro eNB coverage area (Figure 4). We simulate the central sector (shadowed in Figure 4) and the rest of macro eNBs emulate the aggregated interference of a larger network. The wireless channel comprises two components: the deterministic pathloss and the stochastic shadow fading.

The aggregated interference at each UE receiver consists of the power received from all interfering eNBs in the sector (pico and macro) plus the interference from the macro eNBs from other sectors. Fig. 5 shows, for one simulation run, the average traffic load following a daily periodic pattern, and the per-stage traffic load randomly generated from this pattern.

A throughput sample is defined, according to the 3GPP guidelines [37], as the quotient between the size of a downloaded file and the time required to download it. Each UE has to download one file, and therefore generates one throughput sample. We consider that a UE fulfills its QoS requirement if its throughput is over \( T_{\text{min}} = 100 \) kbps. In our simulations, the QoS function \( Q \) provides the expected ratio of UEs satisfying \( T_{\text{min}} \). The power consumption model is defined in Section III-B and its parameter configuration [35] is shown in Table 3.

We consider an horizon of \( N = 24 \) stages, corresponding to the hours of the day. For each stage, we perform 60 evaluations of 100 LTE frames with varying traffic corresponding to each minute within the hour. At each minute, we generate the traffic load according to a Poisson process whose mean is given by the average traffic profile shown in Fig. 5. Each UE is dropped uniformly over the macro eNB coverage area.
<table>
<thead>
<tr>
<th>TABLE 3. Simulation parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network layout</strong></td>
</tr>
<tr>
<td><strong>System bandwidth</strong></td>
</tr>
<tr>
<td><strong>LTE frame duration</strong></td>
</tr>
<tr>
<td><strong>Transmit power</strong></td>
</tr>
<tr>
<td><strong>Macro sector antenna pattern</strong></td>
</tr>
<tr>
<td><strong>Pico antenna pattern</strong></td>
</tr>
<tr>
<td><strong>Antenna gains</strong></td>
</tr>
<tr>
<td><strong>Macro to UE path loss</strong></td>
</tr>
<tr>
<td><strong>Pico to UE path loss</strong></td>
</tr>
<tr>
<td><strong>Shadow fading</strong></td>
</tr>
<tr>
<td><strong>Thermal noise</strong></td>
</tr>
<tr>
<td><strong>Scheduling algorithm</strong></td>
</tr>
<tr>
<td><strong>Traffic model</strong></td>
</tr>
<tr>
<td><strong>File size</strong></td>
</tr>
<tr>
<td><strong>Minimum distances</strong></td>
</tr>
<tr>
<td><strong>Macro consumption parameters</strong></td>
</tr>
<tr>
<td><strong>Pico consumption parameters</strong></td>
</tr>
</tbody>
</table>

**FIGURE 6.** Performance of the switching on-off policies in terms of QoS for different numbers of active pico eNBs.

**FIGURE 7.** Performance of the switching on-off policies in terms of 5th percentile throughput for different numbers of active pico eNBs.

with probability $\frac{1}{3}$, or over a pico eNB coverage area with probability $\frac{2}{3}$.

Regarding the interference coordination controls, we consider $\Gamma = \{0, \frac{1}{3}, \frac{2}{3}, \ldots, \frac{2}{5}\}$ and $\Phi = \{0, 6, 9, 12, 18\}$. The number of pico eNBs per sector is $P = 6$. The rest of the simulation parameters are shown in Table 3.

### B. EVALUATION OF CSA STRATEGY

In this subsection we evaluate the losses associated with the state/control space transformation. In order to revert the transformation, the eNBs with lower adjacency values are switched off firstly, as explained in Sec. V-B. We compare the proposed strategy with other alternative strategies, including the optimal one. We consider for this evaluation $P = 6$, $\gamma = 6/8$ and $\phi = 6$ dBs.

Figures 6 and 7 show the performance in terms of QoS and 5th percentile throughput [37] of the following policies:

- Cell adjacency order: eNBs with lower values of cell adjacency are deactivated firstly.
- Cell adjacency reverse order: eNBs with higher values of cell adjacency are deactivated firstly.
- Optimum case: We select the best of the $P!$ possible orderings of $P$ pico eNBs in the sector.
- Worst case: We select the worst of the $P!$ possible orderings of $P$ pico eNBs in the sector.

Both figures show that Cell Adjacency Order performs similarly to the optimal policy. We performed the simulations of our proposed policy for different values of the weighting parameter $w$ ranging from 0 to 1. The best performing value was 0.4, which is the one used in the following results.
**C. EVALUATION OF ESTIMATORS PERFORMANCE**

The estimators $\hat{C}$ and $\hat{Q}$ are obtained in the offline phase. These estimators are used firstly in the computation of the cost-to-go function, and then in the control selection during the online operation (11). The error incurred by these estimators may affect the final performance of our ADP algorithm. For example, if a control $u'$ and a traffic $\lambda$ are misclassified as meeting the QoS constraint ($Q(\lambda, u') = 1$), the control $u'$ could be selected under traffic $\lambda$, resulting in a non-fulfillment of the QoS requirement in this stage.

We evaluate the errors of the estimators against the available data in the offline phase. That is, we have computed the estimators $\hat{C}$ and $\hat{Q}$ for training sets of different sizes, ranging from 50 to 23,000 samples. For all the obtained estimators, their errors are estimated using the same test set. This way, we obtain better quality estimators as the size of the training set increases. For the polynomial regressor $\hat{C}$, we evaluate the RMS error, and for the classifier $\hat{Q}$, we evaluate the accuracy (ratio of correct classifications), and the ratio of type I and type II errors. Table 4 summarizes the results of these evaluations and the performance of the proposed ADP algorithm in terms of daily energy consumption and QoS satisfaction when using these estimators.

**D. COMPARATIVE EVALUATION OF THE ADP ALGORITHM**

In our numerical evaluation we compare the following policies:

- Our proposed ADP algorithm controlling the Energy Saving (ES) and the Interference Coordination (IC) mechanisms (ADP ESIC).
- Our proposal, but only controlling the ES mechanism (ADP ES).
- A default network configuration where ES and IC mechanisms are not operative.
- An Oracle policy which minimizes the overall daily consumption satisfying the QoS requirement at every stage.

This policy has been computed by means of exhaustive search over the possible controls at each stage.

- An Oracle policy (Oracle ES) which, as ADP ES, only controls the ES mechanism. This policy is an upper bound of previous works that only consider the ES scheme.

Fig. 8 shows the controls selected by each policy at each hour (decision stage) for a threshold $Q_{min} = 0.6$. Fig. 9 shows the aggregated power consumption at each stage, i.e., the consumption of all the eNBs within the sector. Fig. 10 shows the QoS observed at each step, i.e., the portion of total UEs in the sector satisfying $T_{min}$.

ADP ESIC uses, in most stages, fewer pico eNBs than ADP ES, as Fig. 8 shows, implying significant energy savings. The use of IC makes it possible to satisfy the QoS threshold using fewer pico eNBs. Specifically, the Oracle, ADP ESIC and ADP ES policies attain an energy saving with respect to the default configuration of 26.79%, 24.07% and 10.96%, respectively.

Note that the QoS threshold balances power consumption and QoS. If we impose a higher QoS threshold, more active
TABLE 4. Performance of estimators and ADP algorithm as a function of the number of samples in the training set.

<table>
<thead>
<tr>
<th>Number of samples</th>
<th>50</th>
<th>300</th>
<th>1000</th>
<th>7000</th>
<th>18000</th>
<th>23000</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE in $\hat{C}$</td>
<td>263.80±59.10</td>
<td>13.08±1.06</td>
<td>3.95±0.14</td>
<td>3.11±0.01</td>
<td>3.07±0.02</td>
<td>3.08±0.03</td>
</tr>
<tr>
<td>Accuracy in $\hat{Q}$</td>
<td>0.83±0.03</td>
<td>0.88±0.01</td>
<td>0.92±0.01</td>
<td>0.95±0.01</td>
<td>0.95±0.01</td>
<td>0.96±0.01</td>
</tr>
<tr>
<td>Type I error in $\hat{Q}$</td>
<td>0.13±0.04</td>
<td>0.09±0.03</td>
<td>0.06±0.02</td>
<td>0.04±0.01</td>
<td>0.04±0.01</td>
<td>0.03±0.01</td>
</tr>
<tr>
<td>Type II error in $\hat{Q}$</td>
<td>0.25±0.09</td>
<td>0.16±0.05</td>
<td>0.13±0.05</td>
<td>0.06±0.02</td>
<td>0.05±0.02</td>
<td>0.07±0.02</td>
</tr>
<tr>
<td>Daily cons. (kWh)</td>
<td>29.95±0.30</td>
<td>28.70±0.03</td>
<td>28.64±0.00</td>
<td>28.64±0.00</td>
<td>28.64±0.00</td>
<td>28.64±0.00</td>
</tr>
<tr>
<td>QoS fulfillment prob.</td>
<td>0.31±0.11</td>
<td>0.54±0.12</td>
<td>0.59±0.11</td>
<td>0.78±0.05</td>
<td>0.85±0.06</td>
<td>0.90±0.05</td>
</tr>
</tbody>
</table>

FIGURE 11. Selected control $\mu'$ at each hour for $Q_{\text{min}} = 0.68$ under the different policies.

FIGURE 12. Comparison of the aggregated instantaneous power consumption at each hour for $Q_{\text{min}} = 0.68$ under different policies.

FIGURE 13. Ratio of UEs satisfying the minimum throughput $T_{\text{min}}$ at each hour for $Q_{\text{min}} = 0.66$.

FIGURE 14. Accumulated energy consumption in one day for different values of $Q_{\text{min}}$.

Pico eNBs are needed and therefore the power consumption increases. We illustrate that in Figures 12, 13 and 11 where $Q_{\text{min}} = 0.68$. In this case, the energy savings of Oracle, ADP, ESIC and ADP ES with respect to the default configuration are 22.54%, 20.43% and 9.84%, respectively. It is worth mentioning that in Fig. 12 the instantaneous power consumption of the Oracle policy exceeds that of our proposal in three stages due to the activation power consumption spikes. However, the Oracle policy attains the minimal overall daily consumption, which is the objective according to Eq. (4).

For both $Q_{\text{min}}$ values our proposal performs very closely to the Oracle policy. This indicates that the potential efficiency losses associated to the approximation strategies of our proposal are not significant, and the performance of our ADP policies are very close to the optimal ones. It is worth mentioning that when IC is not considered (ADP ES and
Default configuration), the QoS threshold cannot be satisfied during the traffic peak between stages 11th and 15th.

Finally, Fig. 14 illustrates the relation between the QoS threshold, $Q_{\text{th}}$, and the best achievable daily energy consumption (fulfilling this threshold). The higher the value of $Q_{\text{th}}$, the more energy is needed to satisfy the QoS requirement. This figure also shows that combining the energy saving and interference coordination mechanisms attains the lowest energy consumption under any QoS requirement.

VII. CONCLUSION

This paper presented a novel stochastic control approach addressing the energy saving and the interference coordination problems jointly. It is applied to an LTE-A HetNets to minimize the energy consumption assuring QoS guarantees. Considering the periodic pattern of daily traffic demands, the problem is formulated as a finite horizon MDP where we apply the CEC principle for the sake of computational tractability. Specifically, we propose an ADP algorithm combining the CEC principle and machine learning techniques. We have shown the considerable improvement associated to addressing energy saving and interference coordination problems jointly. In addition, our proposal allows a configurable QoS requirement.

There are, however, two aspects left for future research: The first one is clustering of sectors explained in Sec. V. The fewer the sectors per cluster, the more similar the traffic profile among clusters, which can increase the performance of our proposal, at the cost of requiring more instances of the ADP controller. On the other hand, larger clusters lead to more variance in the traffic intensity of the sectors. This trade-off has to be further investigated. Second, we assume a daily periodic pattern in the traffic demands which can change depending on the day. For example, the traffic demands can change on weekdays or during special events like sport games or concerts. In the case of regular weekends or vacation days the prediction is straightforward. However, for special events the traffic demands can change depending on the amount of people, the profile of these people, the type of the event, etc. In order to address this issue, we can use external information (e.g. event program in the city) or detect the changes in the traffic patterns in an online fashion and adapt the algorithm to them.

REFERENCES


JOSE A. AYALA-ROMERO received the B.Sc. degree in telecommunications engineering and the M.Sc. degree in telecommunication engineering from the Technical University of Cartagena, Cartagena, Spain, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Department of Information and Communications Technologies. His research interests are cellular networks and learning algorithms.

JUAN J. ALCARAZ received the Engineering degree from the Polytechnical University of Valencia in 1999 and the Ph.D. degree from the Technical University of Cartagena (UPCT) in 2007. After working for several telecommunication companies, he joined UPCT in 2004. He was a Fulbright Visiting Scholar at the Electrical Engineering Department, UCLA, in 2013, and a Visiting Professor with the Department of Information Engineering, University of Padova, in 2017. He is currently an Associate Professor with the Department of Information and Communication Technologies, UPCT. 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, and 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.

* * *
Contextual Bandit Approach for Energy Saving and Interference Coordination in HetNets

Jose A. Ayala-Romero*, Juan J. Alcaraz*, Andrea Zanella†, Michele Zorzi†
*Department of Information and Communications Technologies, Technical University of Cartagena, Spain
†Department of Information Engineering, University of Padova, Italy
email: josea.ayala@upct.es, juan.alcaraz@upct.es, zanella@dei.unipd.it, zorzi@dei.unipd.it

Abstract—This paper addresses the joint problem of energy saving and interference coordination in heterogeneous networks (HetNets) using a contextual bandit formulation. We propose a semi-distributed scheme consisting of a learning agent and local controllers. The learning agent comprises a neural network (NN) classifier and a Multi-Armed Bandit (MAB) algorithm. The NN classifier is dynamically trained to choose a subset of configurations (i.e., feasible configurations in terms of QoS) based on the context information (network state). Then, the MAB algorithm picks one control (i.e., global configuration parameters) among those selected by the NN classifier, with the aim of improving the energy efficiency. These global configurations are interpreted by the local controllers on each network sector. This scheme allows the learning agent to progressively learn the best policy by observing the network state and the performance of the chosen configurations in terms of energy consumption and QoS. Our numerical results show an energy saving close to 20% with respect to a default policy and an improvement of 13% with respect to addressing energy saving and interference coordination separately.

I. INTRODUCTION

A promising step towards increasing the network capacity is based on the dense deployment of small cells, thus realizing the so-called Heterogeneous Networks (HetNets), considered as one of the key technologies in 5G systems [1]. Nevertheless, the densification of HetNets poses two main challenges: the increment of the energy consumption due to the increase in the number of cells, and the higher inter-cell interference. These two issues are intertwined, in that the energy saving mechanisms can affect the inter-cell interference (e.g., when switching cells on and off) and, on the other hand, the interference coordination mechanisms determine the utilization of the radio resources in the different cells, with an impact on their energy consumption. Nonetheless, so far these two issues have been addressed separately.

In contrast, this paper proposes a learning framework that jointly considers the energy efficiency and the interference coordination functionalities. Figure 1 provides a high-level description of our proposal. Its main elements are: the controlled HetNet comprising \( m \) sectors, a central entity (Global Controller) and \( m \) Local Controllers (LC), one per sector. The Global Controller receives performance metrics and context data from the network and makes global control decisions based on this information. The global control, which comprises both interference and energy saving decisions, is broadcast to the Local Controllers. Each Local Controller decides how to effectively translate the global control into a local decision based on the specific pico eNodeB (eNB) deployment within its sector. The main feature of the Global Controller is that it is capable of learning from the observed performance of past decisions to gradually approach the most efficient operation policy. This learning process is performed by a novel algorithm based on a contextual bandit approach [2]. The main challenge addressed by our proposal is to minimize the performance losses associated to the learning process. This is accomplished by means of two strategies: first, the use of two decision levels (global and local) which notably reduces the dimension of the control space at the Global Controller, and second, the introduction of a classification algorithm for discarding the controls that are considered unable to meet the QoS requirements.

We consider a HetNet in which each macro cell contains multiple pico eNBs in each sector. The pico eNBs can be in active (on) or sleeping mode (off), while the macro eNBs are always active. We apply the enhanced Inter Cell Interference Coordination (eICIC) mechanism proposed by the 3GPP for LTE-A Networks, which will be described in more detail in Sec. II-A. The eICIC parameters are the Cell Range Expansion (CRE) bias and the Almost Blank Subframe (ABS) ratio.

Some previous works have addressed the problem of energy efficiency in HetNets. One of the most usual approaches is to formulate the problem as a Markov Decision Process (MDP) [3], [4]. The inherent computational complexity of the MDPs implies the use of approximate dynamic programming approaches such as Reinforcement Learning (RL) [4]. The complexity and scalability of a learning approach is directly related with the dimensions of the state and control spaces.

![Figure 1. System scheme comprising the Global Controller that learns efficient global controls from the network state and performance metrics, and \( m \) Local Controllers which compute local controls for each sector.](image-url)
(curse of dimensionality). Our proposal overcomes this issue by applying two strategies: 1) using two decision levels (global and local), which makes the dimension of the control space independent of the number of macro sectors, and 2) projecting the controls on a low-dimensional space, which makes the control space scale linearly with the number of pico eNBs, instead of exponentially.

Other works address the problem of interference coordination in HetNets using learning algorithms [5], [6]. For example, the authors in [6] propose Q-learning algorithms for learning ABS ratio and CRE bias. However, to the best of our knowledge, our work is the first that jointly addresses energy saving and interference coordination in HetNets. In contrast to approaches relying on mathematical models of the network [7], our proposal is model-free, i.e., it operates without making any assumption about the network [8]. The main contributions of our proposal are summarized as follows:

- To jointly control the energy saving and the interference coordination mechanisms in HetNets with QoS guarantees.
- To learn efficient configurations during network operation thanks to a novel contextual bandit approach that combines a Multi-Armed Bandit algorithm with a Neural Network classifier.

We evaluate our proposal in an LTE-A network simulator following the 3GPP guidelines [9]. The next section details the interference coordination mechanism and the consumption model associated to this technology.

II. INTERFERENCE MANAGEMENT AND POWER CONSUMPTION MODEL

A. Interference Management in LTE-A: eICIC

eICIC is an interference coordination mechanism for heterogeneous networks defined in 3GPP Release 10 (LTE-A). To minimize inter-cell interference, the eICIC schedules the radio resources for pico and macro eNBs in different time periods (subframes). It comprises two main features: Cell Range Expansion (CRE) and Almost Blank Subframe (ABS).

The CRE increases the pico eNBs footprint by adding a bias to their received signal reference power. It is intended to balance the offloading (from macro to pico eNBs) in the network. However, the User Equipments (UEs) located in the extended region (CRE region) will generally have a poor channel quality due to the high interference received from the macro eNB. ABS is motivated by the need to improve the performance of UEs located in CRE regions and consists of reserving certain subframes for pico cell traffic only, muting the macro eNBs in those resources (Almost Blank Subframes). The ABS ratio defines the portion of muted subframes over the total number of subframes (muted and not). We consider synchronized muting, as recommended by the 3GPP [10]. That is, the eICIC controls are applied globally to a cluster of macro eNBs with homogeneous traffic profile.

B. eNB Power Consumption Model

The eNB consumption model used in this work is based on 3GPP guidelines [11]. The power consumption of some of the components of an eNB depends on its load. Thus, it is common to assume a linear relationship between RF output power and power consumption of eNB transceivers (TRXs) [11]. The power consumption model of a pico eNB $j$ is given by:

$$C^j_p = e^j \cdot N_{TRX} \cdot (P_0 + R^{j} \cdot P_{\text{max}}) + (1 - e^j) \cdot N_{TRX} \cdot P_{\text{deep}} + \Delta$$

$$\Delta = \begin{cases} 
\beta \cdot P_0 & \text{when eNB } j \text{ is switched on} \\
0 & \text{otherwise} 
\end{cases}$$

where:

- $e^j = 1$ when the pico eNB $j$ is active and $e^j = 0$ otherwise.
- $N_{TRX}$ is the number of TRXs.
- $P_{\text{max}}$ is the maximum RF output.
- $R^{j}$ is the load factor of the pico eNB $j$ and depends on the ABS ratio, the CRE bias, the traffic intensity and the location of UEs.
- $P_{\text{deep}}$ is the power consumption of TRX components in sleep mode.
- $\beta$ is the portion of $P_0$ needed to switch on the pico eNB TRXs.

Note that $\Delta$ captures the consumption associated to switching on a sleeping pico eNB. The power consumption of the macro eNB $i$ is given by

$$C^i_m = N_{TRX} \cdot (P_{0}^{0} + R^{j} \cdot P_{\text{max}}^{0}) \cdot (1 - \gamma) + N_{TRX} \cdot P_{\text{deep}}^{0} \cdot \gamma$$

where $\gamma$ denotes the ABS ratio. Given the influence of the ABS ratio and the CRE bias on $C^j_p$ and $C^i_m$, our proposal includes these parameters in the control of energy consumption, as explained below.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

We consider an LTE-A access network comprising $M$ sectorized macro eNBs denoted by the set $M$ and $P$ pico eNBs for each macro sector. Let $\mathcal{P}^m$ be the set of pico eNBs overlapping the macro sector $m$, and $\mathcal{P} = \{P^1, P^2, \ldots, P^M\}$ the set of all pico eNBs in the network. We denote the ABS ratio and the CRE bias by $\gamma \in \Gamma$ and $\phi \in \Phi$, respectively, where $\Gamma$ and $\Phi$ are the finite sets comprising all available configurations for these parameters. The time is divided into time stages denoted by $k \in \{0, 1, \ldots\}$.

1) States: Let $e^j_k$ be the state of the pico eNB $j \in \mathcal{P}^m$ at stage $k$, where $e^j_k = 1$ when the pico eNB is switched on and $e^j_k = 0$ otherwise. Let $p^{m}_{k} = \{p^j_k\}_{j \in \mathcal{P}^m} \in \mathcal{E}^m \subseteq \{0, 1\}^P$ be the vector indicating the on/off state of all the pico eNBs in $\mathcal{P}^m$. Let $p_k = \{p^1_k, \ldots, p^M_k\} \in \mathcal{E}$ be the state of all pico eNBs in the network at stage $k$ where $\mathcal{E} = \{0, 1\}^P$. Let $A_{k}$ denote the aggregate traffic load in the network at stage $k$, where $A = [0, A_{\text{max}}]$. We define the system state at stage $k$ as $s_k = (A_{k}, p_{k-1}) \in \mathcal{X}$, where $\mathcal{X} = \Lambda \times \mathcal{E}$ is the state space.

2) Controls: We denote $u_k = (p_k, \gamma_k, \phi_k) \in U$ as the network control, where $U = \mathcal{E} \times \Gamma \times \Phi$ is the control space.
Given $x_k$, the decision maker selects a control $u_k$ based on its previous knowledge. The next state $x_{k+1} = (A_{k+1}, p_k)$ depends on the current control and the traffic at the next stage, which is unknown in advance.

3) Feedback functions: We define two feedback functions: $C: X \times U \rightarrow \mathbb{R}$ representing the aggregated power consumption of macro and pico eNBs in the network (using the model in Sec. II-B) and $Q : X \times U \rightarrow [0, 1]$ which gives the ratio of UEs in the network with satisfactory QoS that, in this study, we consider as a minimum guaranteed throughout. Thus, the feedback function $Q$ indicates the portion of UEs with throughput above a selected threshold. Note that the values obtained from the feedback functions $C$ and $Q$ are random variables due to the randomness of UE locations and traffic demands.

B. Contextual Bandit Problem Formulation

We define a policy as a function $\pi : X \rightarrow U$ which maps states into controls. A learning agent following a policy $\pi$ operates as follows: (i) The learning agent obtains the network state $x_k$ at stage $k$ and selects the control $u_k$ that policy $\pi$ prescribes for $x_k$. (ii) The network operates according to the control $u_k$ during stage $k$, gathering performance measures from each eNB in order to obtain the feedback values ($C_k$, $Q_k$) that are sent back to the learning agent at the end of stage $k$. (iii) The learning agent receives the feedback and updates the policy $\pi$ accordingly.

Our goal is to learn, stage by stage, a policy $\pi$ satisfying our dual objective: to minimize the power consumption while satisfying the QoS requirement. We define the following cost function capturing the tradeoff between these two objectives:

$$\rho(x, u) = C(x, u) + \delta \cdot \max(0, Q \min - Q(x, u)).$$  \hspace{1cm} (4)

The first term accounts for the consumption in the network, and the second one is a penalty that is applied whenever the QoS threshold ($Q \min$) is not satisfied. The coefficient $\delta$ is a weighting factor that regulates the importance of this penalty.

We define the optimal policy $\pi^*$ as the one that minimizes the average cost per stage in the long term, i.e.,

$$\pi^* = \arg\min_{\pi \in \Pi} \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \rho(x_k, \pi(x_k)).$$  \hspace{1cm} (5)

where $\Pi$ is the set of all possible policies. The pseudo-regret (referred to as regret henceforth) of a policy $\pi$ over $N$ stages is given by

$$R_N(\pi) = \sum_{k=0}^{N} E[\rho(x_k, \pi^*(x_k))] - E[\rho(x_k, \pi(x_k))].$$  \hspace{1cm} (6)

This metric accumulates the loss incurred when selecting a suboptimal control at each stage and can then be used to assess the performance of a policy. That is, the lower the regret of a policy, the closer the policy to the optimal one.

However, in order to find policies minimizing (6), it is necessary to deal with the curse of dimensionality since the dimensions of the state and control spaces ($X$ and $U$) make the problem unmanageable as the network size grows. In particular, the size of the set $E$ grows exponentially with the number of pico eNBs per sector and with the number of sectors (|$E$| = $2^P \cdot M$). In this section we present our proposal aimed at addressing these problems.

IV. CONTEXTUAL BANDIT PROPOSAL

As described in the introduction and in Fig. 1, the Global Controller (ClassMAB) is in charge of the learning and control functions. Fig. 2 shows the internal structure of ClassMAB, which comprises two elements, a QoS Classifier and Multi-Armed Bandit modules. ClassMAB and Local Controllers are described in this section.

Let us first introduce some notation regarding the global states and controls. We define $p' = \sum_{i \in U} p^m \in \mathcal{E}' = \{0, \ldots, P\}$ as the projection of the parameter $p^m$, i.e., the total number of active pico eNBs in every sector. We define $x' = (\lambda, p') \in X'$ and $u' = (p', \gamma, \phi) \in U'$ as the global state and the global control, respectively. Note that the dimensions of the global state and control spaces ($X'$ and $U'$) are now linear with $P$ and independent of $M$. The use of global controls is a reasonable assumption when all sectors in $M$ are selected with homogeneous traffic profile, e.g., city center, outskirts of the city, business area, etc. The data flow of our proposal is detailed as follows (Fig. 2):

- The classifier receives the global network state $x'_k$ at stage $k$. Then, it computes the set $U'_{k+1}$ of available controls satisfying the QoS, where $U'_{k+1} = \{u' \in U' : E[Q(x_k, u')] > Q \min\}$. The set $U'_{k}$ is sent to the MAB.
- The MAB selects a control $u' \in U'_{k}$ trying to minimize the energy consumption. This control is sent to the LCs.
- Each LC receives the global control $u'_m$ and generates a local control $u^m$ for its respective sector $m \in M$.
- Each sector $m \in M$ operates with its corresponding local control $u^m$ during stage $k$. At the end of the stage, ClassMAB obtains a global feedback computed from the information provided by the eNBs of each sector. Specifically, the classifier and the MAB obtain a QoS ($Q_k$) and a consumption ($C_k$) measure, respectively.
- Using this feedback, ClassMAB updates its knowledge for the next iteration.

The following subsections explain in detail each module of our proposal.

\footnote{Note that $p'$ is a global network parameter, i.e., the projection of $p^m$ gives the same value of $p'$ for all $m \in M$.}
A. QoS Classifier

The QoS Classifier denoted by $\Theta$ generates at each stage $k$ the set of controls $U'_x \subset U'$ given the state $x'_k$, i.e.:

$$U'_x = \{ u' \in U' : \Theta(x'_k, u') = 1 \}$$

(7)

where $\Theta$ is the classifier’s outcome function which equals 1 if control $u'$ is expected to satisfy the QoS given the traffic intensity $\lambda_k$ (extracted from the global network state $x'_k$), and 0 otherwise.

The classifier is based on a neural network (NN) composed of two hidden layers of 50 nodes each. A rectified linear unit (ReLU) is used as a nonlinear activation function in both hidden layers, and a sigmoid activation function is used in the output layer. The NN receives at each stage $k$ the feedback $Q_k$ from the network. Using this feedback, the NN is trained using the gradient-based algorithm in [12] to minimize the cross-entropy cost function.

B. Multi-Armed Bandit with Constrained Control Availability

The MAB selects a control $u'_k$ from the set $U'_x$ provided by the QoS Classifier at each stage. Classical MAB algorithms [13] assume a fixed set of controls at every stage which makes these algorithms unsuitable for this setting. We propose for this task a modification of the descent $\varepsilon$-greedy MAB algorithm referred to as Control Constrained Descent $\varepsilon$-greedy (CCD $\varepsilon$-greedy) algorithm, shown in Algorithm 1.

Algorithm 1 CCD $\varepsilon$-greedy algorithm

1: Input parameters: $\varepsilon_0$
2: for each stage $k$ do
3: Generate the set $U'_k \subset U'_x$ of unexplored controls
4: if $U'_k$ is empty then
5: if random number $\in [0,1] < \varepsilon_0/k$ then
6: Randomly and uniformly pick one control $u'$ from the set $U'_k$
7: else
8: Select the control $u' \in U'_k$ with minimal expected reward
9: end if
10: else
11: Randomly and uniformly pick one control $u'$ from the set $U'_x$
12: end if
13: Obtain the consumption sample $C_k$ of the configuration $u'_k$
14: Update the expected reward of $u'_k$ with $C_k$
15: end for

C. Local Controllers (LCs)

Each Local Controller $m \in \mathcal{M}$ ($\mathcal{L}C^m$) generates a control $u'^m_k$ from the global control $u'_k$ at each stage $k$. As we discussed in Section II-A, the eICIC controls are inherently global, and therefore, the task of LCs is focused on the projection $E' \rightarrow E$.

We define the cell adjacency value $d^j$ of a pico eNB $j \in \mathcal{P}^m$ in sector $m$ as the weighted average of the distances to the remaining eNBs in the sector:

$$d^j = w \cdot d^m_i + (1-w) \cdot d^p_j$$

(8)

where $d^m_i$ is the distance to macro eNB $m$, $d^p_j$ is the average distance to the other pico eNBs in sector $m$ and $w \in [0,1]$ is a weighting factor which, in this study, is set to $w = 0.4$.

Let $\mathcal{D}^m = \{d^j\} \in \mathcal{P}^m$ denote the set containing the cell adjacency values of all pico eNBs in $\mathcal{P}^m$. Our strategy consists of switching off first the pico eNBs with lower values of cell adjacency since these eNBs can offload their traffic to near eNBs with less degradation in the channel quality.

The $\mathcal{L}C^m$ can compute the values of each element $e^j$ of $\mathcal{D}^m$ from the control $p'$ as follows:

$$e^j = \begin{cases} 1 & \text{if } i \in I \text{ where } I = \mathcal{E}_p(D^m) \\ 0 & \text{otherwise} \end{cases}$$

(9)

where the operator $\mathcal{E}_p(D^m)$ gives us the indices of the $p'$ greatest values in $D^m$, that is, the indices of the $p'$ pico eNBs with the greatest values of cell adjacency.

V. NUMERICAL RESULTS

A. Description of the Simulation Framework

The simulation framework is based on 3GPP guidelines for the evaluation of LTE networks [9]. The network layout comprises 5 sectorized macro eNBs (120 degrees) and several pico eNBs overlapping with each macro coverage area. We simulate the central sector using the remaining sectors to emulate the aggregated interference of a larger network. The wireless channel is composed of the pathloss and the stochastic shadow fading. The aggregated interference at each UE receiver consists of the power received from all interfering eNBs in the sector (picos and macro) plus the interference from the macro eNBs from other sectors.

Each incoming UE generates one throughput measurement, which is defined according to the 3GPP guidelines [9]. A UE satisfies the QoS requirement if its throughput is above $T_{\text{min}} = 100$ kbps. The QoS function $Q$ gives the ratio of UEs satisfying $T_{\text{min}}$. The power consumption model is defined in Section II-B and the values of its parameters are shown in Table I.

The number of pico eNBs per sector is $P = 6$. The sets of available configurations of eICIC parameters are $\Gamma = \{0, \frac{1}{3}, \frac{2}{3}, \ldots, \frac{7}{3}\}$ and $\Phi = \{0, 6, 9, 12, 18\}$. The remaining simulation parameters are shown in Table I. The simulation framework has been developed using Python. The neural networks are implemented using TensorFlow.

B. Simulation results

In this section, we provide numerical results for our proposal and compare its performance with the following benchmarks:

- Oracle provides an upper bound in terms of regret. It selects, for each network state $x_k$ at stage $k$, the optimal control $u'_k \in U'$, which is found by exhaustive search.
- Default configuration is a fixed control where energy saving and interference mechanisms are deactivated.
- NeuralBandit [14] implements a contextual bandit algorithm based on neural networks. It is aimed at learning the cost function $p'(x', u')$ for each arm $u'$ given the context $x'$. It comprises, for each arm, a neural network composed...
of two hidden layers with 20 nodes each. At each stage, an arm is selected according to an ε-greedy policy with decreasing ε. Then, the selected arm is trained using [12] with the feedback measures $C_k$ and $Q_k$.

- **ClassMAB (ES)** evaluates our proposal only controlling the energy saving mechanism, i.e., $γ = φ = 0$.

Other state-of-the-art contextual bandits algorithms, e.g. [2], assume that the expected value of each arm is linear with respect to the context. However, our cost function (4) shows a threshold structure, making these algorithms unsuitable for this application.

Our simulations are aimed at assessing the performance of the algorithms in two phases: a training phase composed of 1200 epochs with 200 stages of variable traffic intensities, and a test phase where the learning state of the algorithms is frozen setting a greedy policy (i.e., using the control with the lowest expected regret). Both **ClassMAB** and **NeuralBandit** start with an initialization period where each arm is selected once and are configured with $ε_0 = 30$. The weighting factor of the cost function is set to $λ = 1 \cdot 10^6$ and $Q_{\text{min}} = 0.6$.

Fig. 3 shows the accumulated regret during the training phase. The regret slope of **ClassMAB** tends to zero at the end of the training phase, i.e., it selects controls very close to the optimum. **NeuralBandit** shows a slower learning rate and a steeper slope in its long term regret. **ClassMAB (ES)** obtains a constant regret slope, which reflects the loss incurred when neglecting the interference coordination mechanism. The regret of the default configuration grows linearly, exceeding that of our proposal at the end of the training period. Fig. 4 shows the value of the cost function at each stage. We can observe the fast convergence of **ClassMAB**, and its ability to operate at smaller cost values compared to other benchmarks.

The test phase was simulated for a one day period, using a stage duration of 10 minutes (a total of 144 stages). At each stage, a random traffic intensity was generated according to the traffic profile shown in Fig. 5. Fig. 6 shows the regret measured during the test phase. After training, **ClassMAB** incurs the lowest regret. **NeuralBandit** obtains the second best result despite its slower learning rate during the training phase. Fig. 7 shows the consumption at each stage. Note that, in general, the consumption pattern resembles the traffic profile (Fig. 5). The crosses on the consumption curves indicate the stages where the selected control does not satisfy the QoS. The benchmarks not using the interference coordination mechanism, besides suffering higher regret, occasionally miss the QoS requirements even using higher consumption controls. This highlights the importance of a synergistic and joint use of interference coordination and energy saving mechanisms. **ClassMAB** shows a power consumption closer to the optimum and always satisfies the QoS requirement. The numerical results of the test phase are summarized in Table II where we show the energy savings with respect to the default configuration and the ratio of stages where QoS has been satisfied.
VI. CONCLUSION

This paper presented a novel contextual bandit approach addressing energy saving and interference coordination in LTE-A HetNets. In our proposal, the learning agent (Global controller) learns efficient global configurations based on network performance samples and context information (global state of the network). These global configurations are interpreted locally by the Local controllers on each network sector. Breaking down the decision into two levels allows the complexity of the control space to grow linearly rather than exponentially, favoring the scalability of our mechanism. Our proposal is a novel combination of a neural network classifier and a multi-armed bandit algorithm which is able to minimize the energy consumption providing QoS guarantees. Our numerical results show an energy saving close to 20% with respect to a fixed default policy. The considerable improvement associated to addressing energy saving and interference coordination problems jointly is also shown.

VII. ACKNOWLEDGEMENTS

This work was supported by project grant AEI/FEDER TEC2016-76465-C2-1-R (AIM). Jose A. Ayala-Romero acknowledges personal grant FPU14/03701.

REFERENCES

Online Learning for Energy Saving and Interference Coordination in HetNets

Jose A. Ayala-Romero, Juan J. Alcaraz, Andrea Zanella, Senior Member, IEEE, and Michele Zorzi, Fellow, IEEE

Abstract—In heterogeneous cellular networks (HetNets), switching off small cells under low user traffic periods has been proved to be an effective energy saving strategy. However, this strategy has strong interactions with interference coordination (IC) mechanisms, making it convenient to address both tasks simultaneously. The motivation of this paper is to develop a self-optimization algorithm capable of jointly controlling energy saving and IC functionalities. Our proposal is based on a contextual bandit formulation that, among other challenges, implies discovering the most energy-efficient control actions while satisfying a predefined level of Quality of Service (QoS) for the users. We propose a two-level framework comprising a global controller, in charge of a group of macro cells, and multiple local controllers, one per macro cell. The global controller implements a novel algorithm, referred to as the Bayesian Response Estimation and Threshold Search (BRETS), that is capable of learning, for each control action, its feasibility boundaries in terms of QoS and its energy consumption as a function of the aggregated user traffic. The algorithm comes with a bound on its expected convergence time. The local controllers translate the control actions learned by the global controller into local decisions. Our numerical results show that BRETS is only 1% less efficient than an ideal oracle policy, clearly outperforming other benchmark algorithms.

Index Terms—Online learning, contextual multi-armed bandit, green networks, heterogeneous networks, interference coordination.

I. INTRODUCTION

A PROMISING step towards increasing the network capacity is based on the dense deployment of small cells, thus realizing the so-called Heterogeneous Networks (HetNets), considered one of the key technologies in 5G networks [1]. Nevertheless, the densification of HetNets poses two main challenges: the increment of the energy consumption due to the larger number of cells, and the inter-cell interference from the macro to the small cells. For the first issue, the most effective energy saving (ES) strategy is to switch off underutilized small cell stations, if this is possible, while avoiding noticeable degradation of the Quality of Service (QoS) of the network users [2]. For the second one, interference coordination (IC) mechanisms allow the network to optimize its wireless access capacity by adjusting the interference level and the radio resource allocation between macro and small cells [3]. However, these two mechanisms are intertwined: switching on/off small cells yields a reassignment of the user equipments (UEs) to the active base stations and, consequently, a change in the inter-cell interference. In turn, the IC control mechanisms will adjust the transmission resources assigned to the different UEs and their transmit power, thus impacting on the overall energy consumption of the system. To exemplify the concept, we report in the upper row of Fig. 1 the heat maps of the overall energy consumption and in the lower row the corresponding heat maps of the fraction of UEs with satisfactory QoS (i.e., sufficiently high throughput in this example), when varying the number of active picocells from 0 to 6 (left to right). For each number of active picos, the heat maps are obtained by changing the configuration of the Almost Blank Subframe (ABS) and the Cell Range Expansion (CRE) bias, which are two IC mechanisms that will be explained in detail later. We can observe how energy efficiency and QoS satisfaction jointly depend on the number of active picos and the setting of the IC parameters. Therefore, addressing both tasks simultaneously has been shown in [4] to improve ES while maintaining a desired level of QoS.

Fig. 1. Energy consumption and QoS satisfaction values in a macro cell for different combinations of IC and ES controls under the same traffic conditions. IC controls are given by the Almost Blank Subframe (ABS) ratio, and Cell Range Expansion (CRE) bias, while ES controls are determined by the number of active picos in the macro cell.

On the other hand, the self-optimization of network control tasks is also considered a key feature of 5G networks [5, 6], enabling the network to autonomously find the most efficient configuration for each functionality without need for human intervention. To attain this objective, an online learning approach is especially interesting since it aims at learning the most effective configurations using observations taken from the real operating network. Offline learning algorithms, in contrast, need to be trained before their implementation in the real system, requiring either a simulated model of the network, or a data set obtained from the system, involving additional costs. Moreover, the resulting control policies would be effective as long as the real network behaves as predicted during the offline training phase. Inaccuracies of the simulation model, biases in the data set, or changes in the network, might reduce the effectiveness of the policy. Notably, although there exist previous online learning proposals for IC [7], [8] and ES [9] separately, this approach has not yet been applied to the joint control of IC and ES, which is the motivation of this work.

Designing an online learning scheme for IC-ES is a challenging task because of several reasons. First, the performance degradation associated to the exploration of poor performing controls should be kept at minimum. Second, the dimension of the control space can be very large because in principle it comprises all the combinations of IC and ES control values. Third, the optimal IC-ES configuration depends on the user traffic intensity in the network, which changes over time. Fourth, our problem involves keeping the QoS perceived by the users above a certain value.

The online learning approach fits a multi-armed bandit (MAB) problem where the state of the system contexts changes independently of the controls, i.e., a contextual bandit problem [10], whose objective is to learn the best configuration at each possible intensity of the network traffic (context). Nevertheless, while classical contextual bandits only consider one performance metric, we must consider two: energy saving and QoS fulfillment. The latter is introduced in the problem as a constraint, resulting in a novel variant referred to as constrained contextual bandit problem. Besides, the contexts take values from a continuous set, unlike the usual case, where a finite set is considered.

Our application scenario is an HetNet composed of a set (or cluster) of contiguous macro eNodeBs (eNBs), and multiple pico eNBs overlapping the coverage area of the macro eNBs. The IC functionality considered is the enhanced Inter Cell Interference Coordination (eICIC) mechanism proposed by the 3GPP for LTE-A Networks [3]. Our framework comprises two decision levels, global and local, corresponding to the cluster and the individual macro eNBs, respectively. At the global level, a centralized entity (global controller) makes IC-ES decisions for the whole cluster and obtains performance observations from the eNBs of the cluster. The online learning algorithm operates in the global controller, allowing it to progressively learn how to select better controls based on the history of past decisions and observations. At the local level, the local controllers decide how to effectively translate each global configuration prescribed by the global controller into a local configuration for each macro cell.

This approach, introduced in our earlier work ClassMAB [11], relies on the results from previous works [7], [12], according to which it is more effective to use the same IC control in sets of contiguous eNBs (synchronized muting), instead of using different controls for each eNB.

However, the learning algorithm proposed in this paper is substantially different from ClassMAB and follows a novel strategy consisting of associating each control action with two concurrent learning processes. The first one, referred to as Threshold Search (TS) is a new mechanism based on the premise that, for each control action, the QoS objective is fulfilled only when the UE traffic intensity is below some threshold. The traffic threshold for each control is initially unknown, and the efficient discovery of all the thresholds is the objective of TS. The second learning process, Response Estimation, aims at estimating the function that maps traffic intensity to network energy consumption (the network response) for each control. Response Estimation uses a Bayesian approach, in particular a Gaussian Process, which generates response function estimations with relatively few samples and provides the uncertainty of the estimation at each traffic intensity. This allows us to apply the principle of optimism in the face of uncertainty used by classic bandit algorithms [13], [14] in a novel way: each control action is associated to a function instead of a scalar. This concurrent learning strategy allows our algorithm to obtain information about multiple context values at each decision stage, increasing its sampling efficiency and therefore its learning rate with respect to other alternatives such as ClassMAB.

In summary, the main contributions of this work are:

- An online learning framework for the joint control of energy saving and interference coordination mechanisms in HetNets based on a new variant of contextual bandit problems which comprises a constraint.
- A novel approach to the above problem based on associating two learning processes to each control, one per performance metric. Each process aims at learning a function mapping the context variable (network traffic intensity) to one performance metric.
- A new algorithm (Threshold Search) implementing the learning process associated to the QoS response of the system. This algorithm is characterized in terms of convergence time.

The remainder of the paper is organized as follows. In Section II the related work and contribution summary are given. In Section III we describe the interference management mechanism and the system model allowing us to formulate the contextual bandit problem. In Section IV we describe our joint coordination framework and our proposed exploration algorithm, which is analyzed in Section V. Finally, the numerical results are given in Section VI and the conclusions are provided in Section VII.

### II. RELATED WORK

Both ES and IC in HetNets have been widely investigated during the last years, generally as separate problems. One of the most usual approaches in ES is to formulate the
problem as a Markov Decision Process (MDP) [28], [29]. The inherent computational complexity of the MDPs implies the use of approximate dynamic programming approaches such as Reinforcement Learning (RL) [28]. The complexity and scalability of a learning approach is directly related to the dimensionality of the state and control spaces (curse of dimensionality). Other works make a compact representation of the state space using a function approximation [28], [32]. Specifically, Comsa et al. [32] and Comsa [33] propose a RL approach to determine the scheduling rules under QoS constraints. But even with this strategy, RL algorithms rely on offline learning before being implemented in a real network. This is because of the slow convergence properties of RL. The main drawback of this strategy is that it requires a very accurate simulation model for each specific network deployment. Our previous work [4] addressed the IC-ES control also from an offline learning perspective, using dynamic programming and certainty equivalence control. In contrast, our current proposal follows an online learning approach and is designed specifically to operate on the real network without previous training.

Other works have addressed IC in HetNets using learning algorithms [6]–[8], [34]. Simsek et al. [34] propose Q-learning algorithms for learning ABS ratio and CRE bias. Our previous works [7] and [8] propose online learning algorithms for IC configuration control. While [8] applies a multi-armed bandit strategy, [7] is based on Response Surface Methodology. However, these proposals are not applicable to the IC-ES problem considered in this paper, because of the diverse new challenges: the higher dimension of the problem, the absence of the main property exploited by [7] and [8] (unimodality of the system response), the presence of a second performance metric in the form of a constraint, and the inclusion of the network traffic intensity (context) in the decision making process.

Virdis et al. [25] and Zheng et al. [31] address the problem of energy saving in HetNets exploiting the ABS configuration and show that the ABS configuration has a significant impact on the power consumption. Nevertheless, eNB on-off switching is not considered.

The problem of eNB on-off switching has been also addressed as an optimization problem [15], [17]–[23], [25]. These problems are usually addressed using iterative algorithms aimed at finding suboptimal solutions since their computational complexity is NP-Hard in most cases. Moreover, the solutions of these problems have to be recomputed whenever the network state (e.g., traffic intensity) changes. In contrast, our proposal learns efficient configurations for any network state. Some of these works [17]–[23] also take into account the user association problem, but none of them considers interference management which, as our work shows, has a notable impact on the performance of on-off switching algorithms in HetNets.

Some works address the eNB switching problem using Stochastic Geometry [24], [26], [27], [35], [36]. Nevertheless, this approach is based on a network model with some simplifications (e.g., eNBs deployed following a Poisson Point Process, path loss as the channel model). In contrast, our proposal is able to learn using real data from the network and does not require any simplification nor assumption in the network model. Table I summarizes the main aspects of the previous works most related to ours.

Other works have used contextual bandit algorithms in cellular networking problems [9], [37], [38]. In [37] an algorithm for content caching based on contextual bandits is proposed. This algorithm learns the context-dependent popularity profiles in order to update the cache content efficiently. In [38] a contextual bandit algorithm addresses the beam alignment problem in millimeter wave systems. Maghsudi and Hossain [9] propose a multi-armed bandit framework for energy-efficient small cell activation but, in contrast to our work, they do not take into account the state of the network (traffic intensity), and do not consider the influence of the interference coordination mechanisms on the global network performance. To the best of our knowledge, our work is the first to apply a contextual bandit formulation for IC-ES control in HetNets.

### III. System Description and Problem Formulation

In this section, we first detail the IC mechanism and the consumption model associated with this technology. Then, we present the system model and formulate the problem addressed.

#### A. Interference Management in LTE-A: eICIC

The 3GPP Release 10 specifies eICIC [3] as the interference coordination mechanism for LTE-A. To minimize inter-cell interference, eICIC schedules the radio resources for pico and macro eNBs in different time periods (subframes). It comprises two main features: Cell Range Expansion (CRE) and Almost Blank Subframe (ABS).

The CRE increases the pico eNBs footprint by adding a bias to their Received Signal Reference Power (RSRP). It is intended to balance the offloading (from macro to pico eNBs) in the network. 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 footprint of the
pico eNBs. However, the UEs located in the extended region (CRE UEs) will generally have a poor channel quality due to the high interference received from the macro eNB. Note that a CRE UE receives a stronger signal from the macro eNB than from the pico eNB to which it is currently associated. ABS is motivated by the need to improve the performance of CRE UEs and consists of reserving certain subframes for pico cell traffic only, muting data transmission from the macro eNB on some radio subframes (Almost Blank Subframes). That is, the eICIC controls are applied globally to a cluster of synchronized muting, as recommended by the 3GPP [12]. The ABS ratio defines the portion of muted subframes over the total number of subframes (muted and not). We consider the total number of subframes (muted and not). We define the set of policies satisfying this QoS requirement as follows
\[ \Pi_{QoS} = \{ \pi \in \Pi : Q(s, \pi(s)) \geq Q_{min} \forall s \in S \} \] (4)

The SINR at UE receiver \( i \) served by eNB \( j \) is given by:
\[ \text{SINR}_i = \frac{P_{tx} \cdot g_i \cdot j}{\sum_{m \in M_i} P_{tx} \cdot m \cdot g_{i,m} + \sum_{p \in P_i} P_{tx} \cdot g_{i,p}} \] (1)
where:
- \( P_{tx} \cdot j \) is the transmission power of eNB \( j \);
- \( g_{i,j} \) is the channel gain between eNB \( j \) and UE \( i \);
- \( M_i \) is the set of interfering macro eNBs of UE \( i \);
- \( P_i \) is the set of interfering pico eNBs of UE \( i \).

Note that in ABS subframes the eNB on some radio subframes (Almost Blank Subframes).

The power consumption model used in this work is based on 3GPP guidelines [39]. The power consumption of some of the components of an eNB depends on its load. Thus, it is common to assume a linear relationship between RF output power and components of an eNB depends on its load. Thus, it is common to assume a linear relationship between RF output power and intensity and the location of UEs; depends on the ABS ratio, the CRE bias, the traffic intensity. We define the network state at stage \( k \) as \( s_k = (\lambda_k, p_{k-1}) \in S \), where \( S = \Lambda \times \mathcal{E} \) is the state space.

2) Controls: The network control is given by \( a_k = (p_k, \gamma_k, \phi_k) \in A \), where \( A = \mathcal{E} \times \Gamma \times \Phi \) is the control space. Let \( a_k^m = (p_k^m, \gamma_k, \phi_k) \) be the local control for the sector \( m \). Given \( s_k \), the decision maker selects a control \( a_k \) based on its previous knowledge. The upcoming network state, \( s_{k+1} = (\lambda_{k+1}, p_k) \), depends on the current control and on the traffic at the next stage, which is unknown in advance.

3) Feedback functions: We define two feedback functions:
- \( C : S \times A \to \mathbb{R} \) providing the aggregated power consumption of macro and pico eNBs in the network and \( Q : S \times A \to [0, 1] \) which gives the ratio of UEs in the network fulfilling a minimum value of a performance metric selected by the operator. Let \( Q_{min} \) be the minimum value of \( Q \) allowed in the network, i.e., the minimum ratio of UEs in the network meeting the performance metric value selected by the operator. Note that the values obtained from \( C \) and \( Q \) are random variables due to the randomness of UE locations and traffic demands.

### D. Constrained Contextual Bandit Formulation

We define a policy as a function \( \pi : S \to A \) which maps network states into controls. A learning agent following a policy \( \pi \) operates as follows: (i) the learning agent obtains the network state \( s_k \) at stage \( k \) and selects the control \( a_k \) that policy \( \pi \) prescribes for \( s_k \); (ii) the network operates according to the control \( a_k \) during stage \( k \), gathering performance measures from each eNB in order to obtain the feedback values (\( C_k, Q_k \)) that are sent back to the learning agent at the end of stage \( k \); (iii) the learning agent receives the feedback and updates the policy \( \pi \) accordingly.

Our goal is to learn, stage by stage, a policy \( \pi \) minimizing the power consumption, while attaining a minimum desired QoS threshold \( Q_{min} \). We define the set of policies satisfying this QoS requirement as follows
\[ \Pi^{QoS} = \{ \pi \in \Pi : Q(s, \pi(s)) \geq Q_{min} \forall s \in S \} \] (4)
where $\Pi$ is the set of all possible policies. The optimal policy $\pi^*$ is the one that minimizes the average consumption per stage in the long term, i.e.,

$$
\pi^* = \arg\min_{\pi \in \Pi} \lim_{N \to \infty} \frac{1}{N} \mathbb{E} \left[ \sum_{k=1}^{N} C(s_k, \pi(s_k)) \right]
$$

(5)

where the expectation is taken with respect to the traffic intensity $(\lambda_k)$ in $s_k$.

Note that $\Pi^{QoS}$ is initially unknown and must also be discovered by the learning algorithm. This implies that the algorithm needs to select, during the learning process, controls $a_k$ that violate the QoS constraint. Therefore, in order to evaluate the efficiency of the learning process, we need to define a per-stage cost function including a penalty term for violations of the QoS constraint. For this purpose we define

$$
\rho(s, a) = C(s, a) + C_{QoS}(s, a)
$$

(6)

where the penalty function $C_{QoS}$ satisfies the following conditions: if $Q(s, a) < Q_{max}$, then $C(s, a) + C_{QoS}(s, a) > C(s, a')$, where $a'$ is any control satisfying the QoS requirement; and if $Q(s, a) \geq Q_{max}$, then $C_{QoS}(s, a) = 0$. Given this definition of $\rho(s, a)$, the optimal policy $\pi^*$ can be alternatively defined as follows

$$
\pi^* = \arg\min_{\pi \in \Pi} \lim_{N \to \infty} \frac{1}{N} \mathbb{E} \left[ \sum_{k=1}^{N} \rho(s_k, \pi(s_k)) \right].
$$

(7)

The pseudo-regret (referred to as regret henceforth) of a policy $\pi$ over $N$ stages is given by

$$
R_{\pi}(N) = \sum_{k=0}^{N} \left( \mathbb{E} [\rho(s_k, \pi(s_k))] - \mathbb{E} [\rho(s_k, \pi^*(s_k))] \right).
$$

(8)

This metric accumulates the loss incurred when selecting a suboptimal control at each stage and can then be used to assess the performance of a policy. That is, the lower the regret of a policy, the closer the policy to the optimal one. In addition, it also characterizes the convergence rate towards the optimal policy. A sub-linear regret implies that the performance of the algorithm converges towards the optimum.

In order to find policies minimizing (8), it is necessary to deal with the curse of dimensionality since the dimensions of the state and control spaces ($\mathcal{S}$ and $\mathcal{A}$) make the problem intractable as the network size grows. In particular, the size of the set $\mathcal{E}$ grows exponentially with the number of pico eNBs per sector and with the number of sectors ($|\mathcal{E}| = 2^{m \cdot \sum_{n=1}^{m} |\mathcal{P}^{(n)}|}$). Therefore, a solution algorithm should incorporate a dimension reduction strategy, as the one explained in next section.

IV. PROPOSED SCHEME

In this section we first describe the general framework in terms of black box functionality and data flow. Then, we detail the proposed algorithm for finding an efficient control policy. Table II summarizes the most relevant parameters of the proposal.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$, $M'$</td>
<td>Number of macro eNBs, set of macro eNBs</td>
</tr>
<tr>
<td>$p^{m}$</td>
<td>Set of pico eNBs in sector $m$</td>
</tr>
<tr>
<td>$\gamma \in \Gamma$, $\phi \in \Phi$</td>
<td>ABS ratio, CRE bias</td>
</tr>
<tr>
<td>$p^{m}$</td>
<td>Activation state of pico eNBs in sector $m$</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>Activation state of pico eNBs in the network</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Traffic intensity</td>
</tr>
<tr>
<td>$s = (\mathcal{A}, p) \in \mathcal{S}$</td>
<td>Network state, network state space</td>
</tr>
<tr>
<td>$a = (\rho, \gamma, \phi, \mathcal{A})$</td>
<td>Control, control space</td>
</tr>
<tr>
<td>$\alpha^n = (\rho^n, \gamma, \phi)$</td>
<td>Local control for sector $m$</td>
</tr>
<tr>
<td>$C$</td>
<td>Consumption feedback function</td>
</tr>
<tr>
<td>$Q$</td>
<td>QoS feedback function</td>
</tr>
<tr>
<td>$r \in \mathcal{R}$</td>
<td>Ratio of active pico eNBs in the sector</td>
</tr>
<tr>
<td>$\bar{s} = (\mathcal{A}, r) \in \bar{\mathcal{S}}$</td>
<td>Global state, global state space</td>
</tr>
<tr>
<td>$u = (\mathcal{A}, r, \phi) \in \mathcal{U}$</td>
<td>Global control, global control space</td>
</tr>
</tbody>
</table>

Fig. 2. ES and IC framework comprising a HetNet with $M$ sectors, the global controller that learns efficient global controls from the network state and performance metrics, and $M$ local controllers ($LC^{(m)}$), which compute local controls for each sector.

A. Two-Level Framework

Previous works [7], [12] have discussed the performance benefits of applying the same eCIC parameter configuration in contiguous sectors (synchronized ABS). Following this approach, we consider that the access network is divided into groups of contiguous macro cells, referred to as clusters, with similar traffic conditions, such that all the sectors of the cluster share the same eCIC configuration values, $\lambda_k$ and $\phi_k$.

The functional elements of the proposed scheme are depicted in Fig. 2 and consist of the controlled cluster comprising $M$ sectors, a central entity (global controller), and $M$ local controllers ($LCs$), one per sector. This scheme decomposes the decision problem into two levels. At the higher level, the global controller operates with state and action spaces with reduced dimensions, while at the lower level, the local controllers translate the global controls sent by the global controller into $M$ local controls $a_k^{(m)}$, with full dimension, that together constitute the complete network control $a_k$.

This dimension reduction is done by replacing the activation state vector $p_k$ by the ratio of active pico eNBs per sector in the HetNet, $r_k \in \mathcal{R}$, where $\mathcal{R}$ is a finite set containing the possible values of $r_k$. The global state is then defined as $\bar{s}_k = (\lambda_k, r_k)$. Therefore, the global control is defined as
the triplet \( u_k = (r_k, \gamma_k, \phi_k) \). Note that the dimension of the state and control spaces, denoted by \( S \) and \( U \), respectively, are linear with \( |R| \) and independent of \( M \). Therefore, the idea is that similar traffic conditions allowing the \( M \) sectors to benefit from using the same eICIC configuration \( \gamma_k \) and \( \phi_k \), also allow them to share the same ratio \( r_k \), provided that each local controller \( LC^m \) applies \( r_k \) efficiently to generate its local activation vector \( p^u_k \).

The parameters determined by the general controller are passed to the different LCs, which in turn must determine the actual eNB activation patterns \( p^u_k \). To this end, the LCs consider the cell adjacency parameter, which measures how much a pico eNB is isolated, i.e., far from other base stations. The idea is that isolated pico eNBs should not be switched off, to avoid that all their UEs get connected to the macro eNB, increasing the energy consumption. Formally, the cell adjacency of the pico eNB \( j \) is defined as

\[
d^i = w \cdot d^i_m + (1 - w) \cdot d^i_p
\]

where \( d^i_m \) is the distance to macro eNB, \( d^i_p \) is the average distance to the rest of pico eNBs in the sector and \( w \in [0, 1] \) is a weighting factor. Then, the \( |r_k \cdot |P^m|| \) pico eNBs with the largest cell adjacency will be activated, while the others will be switched off. Formally, the elements in \( p^u_k = (e^1_k, \ldots, e^{|P^m|}_k) \) are obtained as follows

\[
e^j_k = \begin{cases} 1 & \text{if } D^m(j) \leq |r_k \cdot |P^m|| \\ 0 & \text{otherwise} \end{cases}
\]

where \( D^m(j) \) is the ranking of pico eNB \( j \) when the pico eNBs are listed in decreasing order of cell adjacency.

To summarize, the proposed framework operates according to the following data flow:

- At each stage \( k \), all macro eNBs receive from their corresponding pico eNBs the information about traffic intensity in their cells. The macro eNB aggregates this information and sends it to the global controller. The global controller computes the overall traffic \( \hat{s}_k \) from the information received from the macro eNBs and determines the general state \( (\lambda_k, r_k) \).
- The global controller selects a global control \( u_k \) and broadcasts it to the LCs.
- Each local controller (\( LC^m \) for \( m \in M \)) receives \( u_k = (r_k, \gamma_k, \phi_k) \) and transforms \( r_k \) into a local vector \( p^u_k \) in order to obtain a local control \( a^m_k = (p^u_k, \gamma_k, \phi_k) \).
- Each sector \( m \in M \) operates with its corresponding local control \( a^m_k \) during stage \( k \).
- At the end of the stage, each macro eNB receives data about the power consumption and the QoS from its corresponding pico eNBs. The macro eNBs aggregates this information to be sent to the global controller. The global controller computes the feedback \( Q_k \) and \( C_k \) from the data received from all the macro eNBs.
- Using this feedback, the global controller updates its knowledge according to the learning process described in the following subsection.

Note that the duration of a stage is determined by the time between consecutive performance observations. Therefore, increasing the signaling frequency implies reducing the duration of a stage, resulting in a tradeoff between signaling overhead and convergence time.

### B. Global Controller Algorithms

In this section we present the Bayesian Response Estimation and Threshold Search (BRETS) algorithm that operates in the global controller selecting the global control \( u_k \) at each stage according to past observations. For each control \( u \in U \), BRETS handles two learning processes in parallel, Threshold Search and Response Estimation, associated to the energy consumption and the QoS fulfillment, respectively. We describe first the principles and strategies used by these processes, and then we will present the complete BRETS algorithm in which both schemes are coordinated.

1) Threshold Search (TS): TS is based on the following premise: each control \( u \) determines the number of pico eNBs in active mode, as well as the interference level and available frame resources for pico and macro UEs (eICIC parameters). Thus, \( u \) is indeed determining the capacity of the controlled network. This implies that, for each \( u \), there exists a user traffic intensity \( \lambda^u_k \) above which the QoS requirement cannot be met. We refer to \( \lambda^u_k \) as the traffic threshold for control \( u \). Only if the traffic intensity is below \( \lambda^u_k \), is the network capacity associated to \( u \) sufficient to attain the required QoS objective.

The value \( \lambda^u_k \) for each control \( u \in U \) is unknown a priori. The objective of TS is to progressively discover the traffic threshold \( \lambda^u_k \) for each control \( u \in U \). Let \( l^u_k \) denote the highest \( \lambda_k \) under which control \( u \) has been used so far, such that the QoS requirement has been satisfied \( (Q_k \geq Q_{\min}) \), and let \( h^u_k \) denote the lowest \( \lambda_k \) for which \( u \) is known not to satisfy the QoS requirement \( (Q_k < Q_{\min}) \). We define the uncertainty region of a control \( u \) at stage \( k \) as the values of \( \lambda \) between the bounds \( l^u_k \) and \( h^u_k \). The threshold \( \lambda^u_k \) is thus contained in this region, i.e., \( l^u_k < \lambda^u_k < h^u_k \). The narrower the uncertainty region, the more accurate the knowledge about \( \lambda^u_k \).

According to the system data flow (Fig. 2), at the end of each stage \( k \) the algorithm receives the feedback measurements \( C_k \) and \( Q_k \) associated with the selected control \( u_k \) and the global state \( \hat{s}_k \). The QoS feedback is used to gradually narrow the uncertainty region of the selected control. Thus, for a selected control \( u \), when \( Q_k \geq Q_{\min} \) (and thus \( \lambda_k \leq \lambda^u_k \)), its lower bound is updated as follows

\[
l^u_k = \max \{\lambda_k, l^u_{k-1}\}
\]

Otherwise \( (\lambda_k > \lambda^u_k) \), the upper bound is updated according to

\[
h^u_k = \min \{\lambda_k, h^u_{k-1}\}
\]

The QoS fulfilling function \( Q \) could comprise multiple QoS constraints, according to the preferences and objectives of the operator. For example, the operator could be interested in guaranteeing that the distribution of the user throughput satisfies a minimum required profile by setting a throughput objective to be attained by at least 5% of its users and another throughput objective for 50% of its users. The traffic threshold for each control will be defined as the maximum network traffic beyond which these two objectives cannot be met using this control. As a consequence, if for a given traffic \( \lambda_k \), a control \( u \) does not satisfy both QoS constraints, then TS updates the upper bound \( h^u_k \) to \( \lambda_k \).
2) Response Estimation: For each control $u$, the function that maps the traffic intensity $\lambda$ to the network energy consumption is referred to as the network response for control $u$. These functions are initially unknown, and must be learned as well, which in our proposal is done by associating each control $u$ to a Gaussian Process (GP) [40].

Let $\mathcal{L}^u$ be a finite set in which we store tuples $(\lambda_k, C_k^u)$ associated to the performance of the control $u$. The maximum length of $\mathcal{L}^u$ is set to $N$. In Bayesian estimation a posterior distribution is computed from a prior distribution and a set of observations. In this work, the GP obtains the posterior of the functions with higher uncertainty $\sigma_{u,\lambda}$ for a control $u$ and a traffic $\lambda$ from a prior distribution and the set of observations $\mathcal{L}^u$.

Let $C_{k}^{u,\lambda} = \mu_{u,\lambda} - 2\sigma_{u,\lambda}$ denote the lower bound of the expected consumption for the control $u$ under the traffic $\lambda$ at stage $k$. The estimation of $C_{k}^{u,\lambda}$ will allow BRETS to find low power consumption controls.

The reasons for using GPs are basically two. First, they require relatively few samples to approximate the functions, because they assume smoothness in the response. In other words, the use of GPs implicitly assumes that a given control obtains similar energy savings for similar traffic intensities, which is reasonable in our setting. Second, GPs provide an estimation of the uncertainty of the estimated response for any traffic intensity. This is especially useful in an online learning framework since the algorithm can exploit the principle of optimism in the face of uncertainty. In our setting, using this principle implies selecting the controls whose estimated energy cost is lowest, considering the lower bound on this estimation $C_{k}^{u,\lambda}$. By doing this, the algorithm tends to explore the areas of the functions with higher uncertainty $\sigma_{u,\lambda}$, but also to exploit the areas with lower empirical average $\mu_{u,\lambda}$. This extends the upper confidence bound (UCB) strategy to the case of learning a function per control action, instead of a scalar.

3) Algorithm Operation: At each stage, the algorithm can be in three different modes of operation: initialization, exploration, and exploitation. The algorithm starts in the initialization mode, which is visited only once. Then, at each stage, the algorithm can enter either the exploration or the exploitation mode. Let us detail each operation mode of the algorithm.

1) Initialization. The algorithm initializes the values of the bounds $l_{k}^{u} = 0$ and $h_{k}^{u} = \lambda_{\text{max}}$ for all $u \in \mathcal{U}$. Then, each $u \in \mathcal{U}$ is selected once. That is, during the first $\lvert \mathcal{U} \rvert$ stages the algorithm explores all controls. After this initial exploration, each control $u \in \mathcal{U}$ has an initial estimation of its average consumption and an initialized lower or upper bound.

2) Exploration. In this operation mode the algorithm aims at reducing the uncertainty regions. To this end, the controller select actions for which the sustainability of the current traffic load $\lambda_k$ is unknown, i.e., actions belonging to the set

$$\mathcal{H}_{k} = \{ u \in \mathcal{U} : l_{k}^{u} < \lambda_k < h_{k}^{u} \}. \quad (11)$$

An example is sketched in Fig. 3, where we can see that the first control satisfies the condition of being in $\mathcal{H}_{k}$.

![Fig. 3. Example of the values of the bounds $l_{k}^{u}$ and $h_{k}^{u}$ for three controls: $\mathcal{U} = \{1, 2, 3\}$. The uncertainty region of each control is marked with a dotted line. Given the traffic intensity $\lambda_k$ shown in the example, control 1 can be selected in the exploration mode; control 2 can be selected in the exploitation mode; control 3 will not be selected since it is expected to fail the QoS requirement $\lambda_k > h_{k}^{3}$.](image)

Let $\mathcal{H}_{k}'$ denote the set of controls $u \in \mathcal{H}_{k}$ for which either $l_{k}^{u} = 0$ or $h_{k}^{u} = \lambda_{\text{max}}$. If $\mathcal{H}_{k}' \neq \emptyset$ then $\mathcal{H}_{k}$ is replaced by $\mathcal{H}_{k}' (\mathcal{H}_{k} \leftarrow \mathcal{H}_{k}')$. This allows the algorithm to continue the initialization of the uncertainty regions in the exploration mode.

To balance exploration and exploitation, we define $B \in \mathbb{N}$ as the exploration budget of the algorithm, which decreases by one unit every time the algorithm enters the exploration mode. Let $\epsilon_0 \in \mathbb{R}^+$ be the exploration constant. A control from $\mathcal{H}_{k}$ is selected if one of the following events takes place: (i) $B > 0$ (ii) $\chi < \epsilon_0/k$, where $\chi \in [0, 1]$ is a uniformly distributed random number. Then, the algorithm picks a control from $\mathcal{H}_{k}$ for which the potential reduction in its uncertainty is the largest:

$$u_k = \arg\max_{u \in \mathcal{H}_{k}} [\min_{(\lambda_k - l_{k}^{u}), (h_{k}^{u} - \lambda_k)}\{ \min_{(\lambda_k - l_{k}^{u}), (h_{k}^{u} - \lambda_k)}\}] . \quad (12)$$

If no exploration control is selected (i.e., $\mathcal{H}_{k} = \emptyset$ or $B = 0$ and $\chi \geq \epsilon_0/k$), the algorithm switches to the exploitation mode.

3) Exploitation. Let $\mathcal{T}_{k}$ be the set containing all exploitation controls (those that are known to satisfy the QoS constraints):

$$\mathcal{T}_{k} = \{ u \in \mathcal{U} : \lambda_k \leq l_{k}^{u} \} \quad (13)$$

For example, the second control in Fig. 3 belongs to $\mathcal{T}_{k}$ and can potentially be selected in the exploration phase. More specifically, the algorithm selects a control from the set $\mathcal{T}_{k}$ such that:

$$u_k = \arg\min_{u \in \mathcal{H}_{k}} C_{k}^{u,\lambda_k}. \quad (14)$$

If $\mathcal{T}_{k} = \emptyset$, the control closest to satisfying the QoS is selected, that is:

$$u_k = \arg\min_{u \in \mathcal{U}} [\lambda_k - l_{k}^{u}]. \quad (15)$$

The operation of BRETS is summarized in Algorithm 1.

V. ALGORITHM ANALYSIS

This section analyzes the exploration strategy (12) of the BRETS algorithm. First we provide a bound on the expected
Algorithm 1 BRETS Algorithm

1: Input parameters: $\varepsilon_0$, $B$
2: Initialization: $l_B^0 = 0$, $h_B^0 = \lambda_{\max}$
3: Pick every control $u \in \mathcal{U}$ once
4: for each stage $k$ do
5: \hspace{1em} $\mathcal{H}_k = \{u \in \mathcal{U} : l_B^u < \lambda_k < h_B^u\}$
6: \hspace{1em} $\mathcal{H}_k' = \{u \in \mathcal{H}_k : l_B^u = 0 \text{ or } h_B^u = \lambda_{\max}\}$
7: \hspace{1em} if $\mathcal{H}_k' = \emptyset$ then
8: \hspace{2em} $\mathcal{H}_k \leftarrow \mathcal{H}_k'$
9: \hspace{end if}
10: \hspace{1em} if $\mathcal{H}_k \neq \emptyset$ and $(B > 0 \text{ or } \chi < \varepsilon_0/k \text{ or } \mathcal{H}_k' \neq \emptyset)$ then
11: \hspace{2em} $u_k = \arg\max_{u \in \mathcal{H}_k} [\mathbb{E}(J^*(0,k) | x_k, \lambda_k, u_k)]$
12: \hspace{2em} $n_k = n_k + 1$
13: \hspace{2em} $\mathcal{H}_k \leftarrow \emptyset$
14: \hspace{2em} \end if
15: \hspace{end if}
16: end if
17: end for
18: Send $u_k$ to LCs
19: Receive the feedback $C_k$ and $Q_k$ from the network
20: Update $\mathcal{L}^u$ with $C_k$ and $Q_k$
21: Update the GP associated with $u_k$ using $\mathcal{L}^u$
22: if $Q_k > Q_{\min}$ then
23: \hspace{1em} $l_B^u = \max \{\lambda_k, l_B^{u_{k-1}}\}$
24: \hspace{1em} $h_B^u = \min \{\lambda_k, h_B^{u_{k-1}}\}$
25: \hspace{1em} end if
26: end if
27: end for

for all $u \in \mathcal{U}$. With a slight abuse of notation we define $|x_k| = (|x_k^1|, \ldots, |x_k^{u_k}|)$, so that the exploration ending condition can be expressed as $|x_k| = (0, \ldots, 0) = 0$. Given a sequence $x_0, x_1, \ldots$, the convergence time is defined as $T = \min\{k : |x_k| = 0\}$.

Definition 1: Let us define as an appropriate exploration strategy any strategy selecting $u \in \mathcal{H}_k$ whenever $\mathcal{H}_k \neq \emptyset$ (as our BRETS strategy does).

The following result provides an upper bound on the expected convergence time of any appropriate exploration strategy.

Lemma 1: Consider that, at every $k$, $\lambda_k$ takes a random value from a finite set $\Lambda$, with a stationary distribution such that $P(\lambda_k = \lambda) > 0$ for all $\lambda \in \Lambda$. Then, the expected convergence time $E[T]$ for any appropriate exploration strategy is bounded as follows

$$E[T] \leq \frac{3}{2 \min_{\lambda \in \Lambda} |\mathcal{H}(\lambda)\|}$$  \hspace{1em} (16)

Proof: See appendix A.

B. Stochastic Shortest Path Model

The exploration process can be modeled as an SSP over an infinite time horizon [41], comprising the following elements:

1) The state of the process at stage $k$ is given by the vector of uncertainty regions $x_k$ and the traffic intensity $\lambda_k$.
2) At each $k$, the exploration strategy $\mu$ (policy in the SSP terminology) determines which $u \in \mathcal{U}$ should be selected according to the observed state $(x_k, \lambda_k)$. We restrict our attention to stationary deterministic policies, i.e., $u_k = \mu(x_k, \lambda_k)$ for all $k = 0, 1, \ldots$
3) When a control $u \in \mathcal{U}$ is selected, $x_{k+1} = x_k'$ for all $u' \neq u$, and $x_{k+1}''$ can take two values: $x_{k+1}'' = x_k$ and $x_{k+1}'' = x_k'$. The first one corresponds to the update $l_B^{u_k+1} = l_B^u$, $h_B^{u_k+1} = \min\{\lambda_k, h_B^u\}$ which occurs with probability $P(\lambda_k \geq \lambda_B^u)$; the second one $x_{k+1}'' = x_k'$ with probability $P(\lambda_k < \lambda_B^u)$.
4) The transition probabilities between consecutive states are determined by the policy $\mu$, and by the probability distributions of $\lambda_k$ and $\lambda_B^u$.
5) The termination state corresponds to $|x_k| = 0$.
6) The per-stage cost is 1 when $|x_k| 
eq 0$ and 0 otherwise. With the above elements we can define the following cost-to-go function for policy $\mu$ at a given state $(x_k, \lambda_k)$, as follows

$$J_{\mu}(x_k, \lambda_k) = \lim_{T \to \infty} \mathbb{E}\left[ \sum_{t=k}^{T} \mathbb{I}_{|x_t| \neq 0} |x_t, \lambda_k\right],$$  \hspace{1em} (17)

where $\mathbb{I}$ is the indicator function, and the conditional expectation is obtained with respect to the transition probabilities induced by $\mu$ and the distributions of $\lambda_k$ and $\lambda_B^u$. The objective is to find, for an initial state $(x_0, \lambda_0)$, the optimal policy $\mu^* = \arg\min_{\mu} J_{\mu}(x_0, \lambda_0)$. At every state $(x_k, \lambda_k)$, the optimal cost-to-go function $J^*$ satisfies Bellman’s equation

$$J^*(x_k, \lambda_k) = 1 + \min_{u \in \mathcal{U}} \mathbb{E}\left[ J^*(x_{k+1}, \lambda_k) | x_k, \lambda_k, u_k = u \right].$$  \hspace{1em} (18)

This ending condition implicitly assumes that $\lambda_B^u > 0$, which is equivalent to assuming that every $u$ fulfills the QoS requirement in the absence of user traffic. Otherwise this $u$ should not even be considered for exploration.
To simplify the notation, we consider only two controls \( U = \{1, 2\} \), although the following discussion can be generalized to any finite set of controls. For \( x_k = (x_k^1, x_k^2) \) we can use the notation \( J^*(x_k^1, x_k^2, \lambda_k) \) or \( J^*(\{k\}, \lambda_k) \) conveniently. Let us develop (18) for \( U = \{1, 2\} \):

\[
J^*(x_k^1, x_k^2, \lambda_k) = 1 + \min \left[ \mathbb{E}\left[ J^*(x_{k+1}^1, x_{k+1}^2, \lambda) \mid x_k, \lambda_k \right] \right]
\]

where

\[
E\left[ J^*(x_{k+1}^1, x_{k+1}^2, \lambda) \mid x_k, \lambda_k \right] = P(\lambda_k \geq \lambda_h^1) \mathbb{E}_\lambda \left[ J^*(x_{k+1}^1, x_{k+1}^2, \lambda) \right] + P(\lambda_k < \lambda_h^1) \mathbb{E}_\lambda \left[ J^*(x_{k+1}^1, x_{k+1}^2, \lambda) \right],
\]

and \( E_\lambda \) denotes the expectation computed with respect to \( \lambda \). Now we provide a definition followed by a result characterizing the structure of the optimal cost-to-go function \( J^* \).

**Definition 2:** Given two uncertainty region vectors \( x_k = (x_k^1, x_k^2) \) and \( z_k = (z_k^1, z_k^2) \), we say that \( x_k \) contains \( z_k \) if \( z_k^1 \subseteq x_k^1 \) and \( z_k^2 \subseteq x_k^2 \), and we express it as \( z_k \subseteq x_k \).

**Lemma 2 (Monotonicity of \( J^* \)):** The optimal cost-to-go function satisfies the following property:

\[
J^*(x_k, \lambda_k) \geq J^*(z_k, \lambda_k), \quad \text{if} \ z_k \subseteq x_k
\]

for all \( \lambda_k \in \Lambda \).

**Proof:** See Appendix B

This monotonicity result can be used to approximately solve the SSP using an index policy, as discussed in the next subsection.

**C. Index Policy**

There are two main difficulties for computing an optimal policy. First, the dimensionality of the problem. Specifically, the state space comprises \( \binom{1+|\Lambda|}{2} \) possible states, which renders the SSP intractable for practical values of \( |\Lambda| \) and \(|\mathcal{U}| \) (note that \( \mathcal{U} \) comprises all possible combinations of the ratio of active stations \( r \), the ABS ratio \( \gamma \), and the CRE value \( \phi \)). Second, the distribution of \( \lambda_h^u \) for each \( u \in \mathcal{U} \) is, in general, not known a priori, which means that the transition probabilities of the SSP cannot be computed accurately.

A feasible approach to overcome the previous limitations is to use an index policy exploiting the structure of the optimal cost-to-go function \( J^* \). According to Lemma 2, for a given \( \lambda \) and \( u \), the reduction in the cost-to-go function, \( J^*(x_k, \lambda) - J^*(x_{k+1}, \lambda) \), is larger for larger reductions of the uncertainty region \( |x_k^u| - |x_{k+1}^u| \). Because there are two possible values of \( x_{k+1}^u \) (\( x_{k+1}^u \) and \( x_{k+1}^u \)) for each control \( u \), one reasonable index policy is to select the \( u \) with the largest expected reduction:

\[
u_k = \arg\max_{u \in \mathcal{U}_k} [P(\lambda_k \geq \lambda_h^u)(h^u_k - \lambda_k) + P(\lambda_k < \lambda_h^u)(\lambda_k - h^u_k)].
\]

Because the distribution of \( \lambda_h^u \) is in general unknown, BRETS implements a worst case approach to (22). Note that (12) is equivalent to

and thus it is a maxmin index policy. Our numerical simulations have shown that the index policy (23) converges almost at the same rate of (22). We evaluated index policies determined by the minimum or the expected largest uncertainty region, and both of them obtained slower convergence rates than (23).

**VI. NUMERICAL RESULTS**

**A. Description of the Simulation Framework**

The numerical evaluations have been performed by using a custom simulation framework developed in Python, which is based on the 3GPP guidelines for the evaluation of LTE networks [42]. The numerical results shown in this section are obtained using synthetic network data generated by this simulator. The network layout comprises 5 sectorized macro eNBs (120 degrees) and several pico eNBs overlapping each macro coverage area. We simulate the central sector using the remaining ones to emulate the aggregated interference of a larger network. The wireless channel is composed of pathloss and stochastic shadow fading. The aggregated interference at each UE receiver consists of the power received from all interfering eNBs in the sector (pico and macro) plus the interference from the macro eNBs from nearby sectors as detailed in (1).

In our numerical evaluations, we define \( Q \) as the ratio of UEs whose throughput is above a minimum value denoted by \( T_{\text{min}} = 100 \text{ bps} \). Each incoming UE generates one throughput measurement, which is defined according to the 3GPP guidelines [42]. The power consumption model is defined in Section III-B and the values of its parameters are shown in Table III.

The number of pico eNBs per sector is \( P = 6 \) and the number of ES controls is \( |\mathcal{R}| = 7 \). The sets of available configurations of eICIC parameters are \( \Gamma = \{0, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, 0\} \) and \( \Phi = \{0, 6, 9, 12, 18\} \). For each control \( u \) one Gaussian process (GP) is maintained by BRETS. For each GP, the maximum length of the set \( \mathcal{L}^u \) is fixed to \( N = 100 \), the mean of the prior distribution is set to zero, and the covariance matrix is specified by the Matern kernel. The hyperparameters of the kernel are optimized using a quasi-Newton algorithm. For the sake of computational efficiency, these hyperparameters are updated (step 20 in Algorithm 1) only 10 times during the training phase of our simulations, whenever the set \( \mathcal{L}^u \) has the length 1, 2, 3, 5, 8, 13, 22, 36, 60, or 100. Gaussian processes have been implemented using the Python toolbox Skict-learn [43]. The remaining simulation parameters are shown in Table III.

**B. Comparison of Local Control Strategies**

In this subsection we compare our proposed local control strategy (eNBs with lower values of cell adjacency are switched off first) to other alternative strategies, including the optimal one obtained by evaluating all possible orderings.
To obtain the numerical results we have set \( P = 6 \), ABS ratio = 6/8 and CRE bias = 6 dB. Fig. 4 and 5 show the network performance in terms of QoS (defined in Sec. III-C) and 5th percentile throughput [42] of the following policies:

- **Proposed order**: eNBs with lower cell adjacency values are switched off first.
- **Reverse order**: eNBs with higher values of cell adjacency are switched off first.
- **Optimum order**: We select the best of the \( P \!) possible orderings of \( P \) pico eNBs in the sector.
- **Worst case order**: We select the worst of the \( P \!) possible orderings of \( P \) pico eNBs in the sector.

Both figures show that the proposed ordering strategy performs similarly to the optimal one. The configuration parameter of the proposed ordering strategy is \( w \in [0, 1] \) which has been extensively evaluated by simulation. We found that its best performing value is \( w = 0.4 \) which has been used in all the numerical experiments of this section.

### C. Benchmark Evaluation

In this section, we provide numerical results for our proposal and compare its performance with the following benchmark algorithms operating in the global controller:

- **Oracle** which selects, for each global state \( \tilde{s}_k \) at stage \( k \), the optimal control \( u^*_k \in \mathcal{U} \), which is found by exhaustive search. Note that the Oracle policy refers to the global controller (\( \mathcal{U} \) is the set of global controls) considering near optimal performance of the local controllers. The performance obtained with the oracle is used for measuring the regret of the algorithms evaluated.
- **Default configuration**, which is a fixed control where energy saving and interference mechanisms are deactivated.
- **NeuralBandit** [44] implements a contextual bandit algorithm based on neural networks. It is aimed at learning the cost function \( \rho(x, u) \) for each control \( u \) (also called arm in MAB terminology) given the context \( x \). It comprises, for each arm, a neural network with two hidden layers of 20 units each. ReLU activation function is considered for hidden layers. At each stage, an arm is selected according to an \( \varepsilon \)-greedy policy with decreasing \( \varepsilon \). Then, the selected arm is trained by means of the optimization algorithm Adam [46], using the feedback measures \( C_k \) and \( Q_k \). That is, each NN is trained with

---

![Fig. 4. Performance of the switching on-off policies in terms of QoS for different numbers of active pico eNBs.](image)

![Fig. 5. Performance of the switching on-off policies in terms of 5th percentile throughput for different numbers of active pico eNBs.](image)

---

**TABLE III SIMULATION PARAMETERS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network layout</td>
<td>5 sectorized macro eNBs, 300 m ISD, ( P = 6 ) pico eNBs per sector</td>
</tr>
<tr>
<td>System bandwidth</td>
<td>10 MHz</td>
</tr>
<tr>
<td>LTE frame duration</td>
<td>Subframe 1 ms, Protected-subframe 8 ms, Frame 10 ms</td>
</tr>
<tr>
<td>Transmit power</td>
<td>Macro eNB 45 dBm, pico eNB 30 dBm</td>
</tr>
<tr>
<td>Macro sector antenna pattern</td>
<td>( A_{\text{ref}} = -\min_{\left{ \frac{\theta}{\pi} \right}} \left{ 12 \left( \frac{\theta}{\pi} \right)^2, A_{\text{max}} \right} ), ( A_{\text{max}} = 70 ) degrees, ( A_{\text{ref}} = 25 ) dB</td>
</tr>
<tr>
<td>Pico antenna pattern</td>
<td>Omnidirectional</td>
</tr>
<tr>
<td>Antenna gains</td>
<td>Macro: 14 dB; pico: 5 dB; UE: 0 dB</td>
</tr>
<tr>
<td>Macro to UE path loss</td>
<td>( 128.1 + 37.6 \cdot \log_{10}(R[Km]) ) where ( R ) is the macro eNB to UE distance</td>
</tr>
<tr>
<td>Pico to UE path loss</td>
<td>( 149.7 + 36.7 \cdot \log_{10}(R[Km]) ) where ( R ) is the pico eNB to UE distance</td>
</tr>
<tr>
<td>Shadow fading</td>
<td>Lognormal distribution with 10 dB standard deviation</td>
</tr>
<tr>
<td>Thermal noise density</td>
<td>-176 dBm/KHz</td>
</tr>
<tr>
<td>Scheduling algorithm</td>
<td>Proportional Fair (PF)</td>
</tr>
<tr>
<td>Traffic model</td>
<td>File Transfer Protocol (FTP)</td>
</tr>
<tr>
<td>File size</td>
<td>0.5 Mbytes</td>
</tr>
<tr>
<td>Minimum distances</td>
<td>Macro - pico: 70 m; Macro - UE: 35 m; Pico - pico: 40 m; Pico - UE: 10 m</td>
</tr>
<tr>
<td>Macro power consumption parameters</td>
<td>( N_{\text{TRX}} = 6, P_{\text{c}} = 130 \text{ W}, P_{\text{m}} = 20 \text{ W} )</td>
</tr>
<tr>
<td>Pico power consumption parameters</td>
<td>( N_{\text{TRX}} = 2, P_{\text{c}} = 56 \text{ W}, P_{\text{m}} = 63 \text{ W}, P_{\text{sleep}} = 39 \text{ W} )</td>
</tr>
</tbody>
</table>
one sample, by means of backpropagation, every time its associated arm is used.

- **ClassMAB** [11] is our previous proposal for the problem addressed. It comprises a neural network classifier aimed at finding controls satisfying the QoS requirement, and a MAB algorithm in charge of selecting controls with low energy consumption. For the MAB algorithm we proposed a modified version ε-greedy in which the set of available controls is variable at each stage.

- **ClassMAB (ES)** refers to the ClassMAB algorithm, but controlling only the energy saving mechanism. This option is included to show the importance of considering interference management when applying energy saving actions.

Other state-of-the-art contextual bandit algorithms, e.g., [10], [14], [47], assume that the expected value of each arm is linear with respect to the context. However, our cost function (6) shows a threshold structure, making these algorithms unsuitable for this application.

Our simulations are aimed at assessing the performance of the algorithms in two phases: a training phase composed of 1200 epochs with 200 stages of variable traffic intensities, and a test phase where the learning state of the algorithms is frozen setting a greedy policy (i.e., using the control with the lowest expected regret). BRETS, ClassMAB and NeuralBandit start with an initialization period where each control is selected once and are configured with ε₀ = 30. The exploration budget of BRETS is set to D = 300. The QoS threshold is set to Q_min = 0.6, and the per-stage cost function is defined as ρ(s, a) = C(s, a) + δ ∗ max(0, Q_min − Q(s, a)), with δ = 10^6. This definition results convenient because of two reasons: first, the penalty imposed is proportional to the QoS degradation, which is useful in terms of benchmark comparison. And second, according to the simulation results, this ρ complies with the conditions in Section III-D, i.e., under the optimal policy (oracle) the penalty factor is never activated since the QoS constraint is always satisfied. All the results presented in this section are the average of 30 independent simulation runs.

Fig. 6 shows the accumulated regret during the training phase. The slope of the regret curve for BRETS is almost flat, indicating that, in the long term, it selects controls very close to the optimal ones. Besides, its accumulated regret is the lowest, compared with the other benchmarks, which indicates that its learning process is the fastest. The regret slope of ClassMAB also approaches zero at the end of the training phase. The accumulated regret of the default configuration is not especially high at the end of the training phase since this policy does not converge to the optimal one.

Figure 7 shows the value of the cost function at each stage. We can observe the fast convergence of BRETS (only a few epochs are needed), and its ability to operate at smaller cost values compared to other benchmarks. Note that, compared to ClassMAB, NeuralBandit shows a slower learning rate during the first 150 epochs, and a higher long-term cost value. Of these two aspects, the first one is the most relevant in the regret curve for NeuralBandit in Fig. 6. The long-term cost value of ClassMAB (ES) is higher than BRETS, ClassMAB and NeuralBandit, which reflects the loss incurred when neglecting the interference coordination mechanism.

We considered a one day period for the test phase, using a stage duration of 10 minutes (i.e., a total of 144 stages). At each stage, a random traffic intensity is generated according to the traffic profile shown in Fig. 8. Note that in the case of special events like sport games or live concerts, the algorithms
could face traffic peak values for which they have not received samples during the training phase. In this situation, BRETS enters the exploration phase (see Algorithm 1) in order to discover which control could better handle this new situation. If the QoS constraints cannot be met, BRETS selects the control closest to this objective. Fig. 9 shows the regret measured during the test phase, during which BRETS obtains the lowest regret, as expected, followed by ClassMAB and NeuralBandit. Because the regret combines energy consumption and QoS fulfillment, it is interesting to evaluate these two metrics separately for comparing performances. Fig. 10 shows the power consumption at each stage. Note that, in general, the power consumption pattern resembles the traffic profile (Fig. 8). It is clear that the power consumption pattern shown by BRETS is the closest to that of the Oracle. Finally, Fig. 11 shows the estimated probability of failing to satisfy the QoS requirement at each stage of the test phase. Note that during the traffic peak (stages between 70 and 80) there are some stages in which the policies neglecting the IC mechanism are not able to meet the QoS requirement using any control, i.e., they fail with probability 1. This shows again the importance of the joint IC-ES control.

The benchmarks that do not incorporate the interference coordination mechanism, besides consuming more energy, are unable to satisfy the QoS requirement in some stages. This highlights the importance of combining interference coordination and energy saving mechanisms. The numerical results of the test phase are summarized in Table IV where we show the energy savings with respect to the default configuration and the estimated probability of failing the QoS.

<table>
<thead>
<tr>
<th></th>
<th>Energy saving (%)</th>
<th>Ratio of QoS fulfillment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>25.70%</td>
<td>1</td>
</tr>
<tr>
<td>Default conf.</td>
<td>0%</td>
<td>0.9752</td>
</tr>
<tr>
<td>NeuralBandit</td>
<td>13.258%</td>
<td>0.9777</td>
</tr>
<tr>
<td>ClassMAB (ES)</td>
<td>8.085%</td>
<td>0.9652</td>
</tr>
<tr>
<td>ClassMAB</td>
<td>19.483%</td>
<td>0.9817</td>
</tr>
<tr>
<td>BRETS</td>
<td>23.007%</td>
<td>1</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

This paper presented an online learning algorithm for jointly controlling the energy saving and interference coordination mechanisms in HetNets. Our framework considers two levels. At the higher level, a global controller selects global controls applicable to a cluster of contiguous macro cells. At the lower level, local controllers decide how to apply the prescribed global control on each macro cell sector. The proposed learning algorithm, running in the global controller, addresses a constrained contextual bandit problem using a novel approach: for each action, the algorithm handles two simultaneous learning processes, one for the performance objective (network energy consumption) and another for the constraint (QoS fulfillment). Each one of these processes aims at learning a function, instead of a scalar as in conventional bandit algorithms. These functions map the context (network traffic intensity) to the performance metric of interest. For the QoS function, we propose a novel strategy for finding the traffic threshold below which each control fulfills the QoS objective. For estimating the energy consumption functions, the algorithm uses Gaussian processes, which allows the BRETS to balance exploration and exploitation when selecting controls. In our numerical simulations BRETS outperformed other alternatives, attaining an energy saving of 25% and
a QoS fulfillment ratio of 100%, very close to the results obtained by an oracle policy.

Future work includes the following lines. Our proposal could be complemented by an auxiliary learning algorithm for finding clusters of contiguous cells in which joint IC-ES controls can be applied effectively, i.e. for assigning cells to global controllers. BRETS instances should be coordinated among contiguous clusters, in order to avoid the mutual influence caused by the interference at their boundaries. A possible approach would be to alternate the learning periods of adjacent clusters. Finally, it would be interesting to evaluate the potential performance improvement of using one ES control per cell instead of a common ES control for the whole cluster. Because a larger control space is associated to a slower learning rate, this might reveal a tradeoff between long-term performance and convergence time.

APPENDIX A
PROOF OF LEMMA 1

The exploration of a control $u$ ends when this control has been used for the traffic values $\lambda^u_0$ and $\lambda^u_H - \delta$. If only one control $u$ has to be explored ($U = \{u\}$), we can obtain, by means of an absorbing Markov chain, the following expected convergence time:

$$E[T^u] = \frac{1}{P(\lambda^u_0) + P(\lambda^u_H - \delta)} \left( 1 + \frac{P(H)}{P(\lambda^u_H - \delta)} + \frac{P(H)}{P(\lambda^u_0)} \right)$$

(24)

where $P(\alpha)$ is short for $P(\lambda_k = \alpha)$. It can be easily checked in (24) that $E[T^u]$ increases when $P(\lambda^u_0) \to 0$, and also when $P(\lambda^u_H - \delta) \to 0$. Since $P_{\min} = \min P(\lambda_k = \lambda)$ is the lower bound for both $P(\lambda^u_0)$ and $P(\lambda^u_H - \delta)$ we have

$$E[T^u] \leq \frac{1}{P_{\min}} \left( 1 + \frac{P_{\min}}{P_{\min}} + \frac{P_{\min}}{P_{\min}} \right) = \frac{3}{2} P_{\min}.$$  

(25)

Consider a (non-appropriate) exploration strategy that explores the controls in $U$ sequentially, selecting current $u$ until $|x^u| = 0$ before starting to explore the next control. Such strategy has an expected convergence time of $\sum_{u \in U} E[T^u]$. Now note that $u \in H_k$, with $H_k \neq \emptyset$, implies that $u$ has not yet been selected under the current traffic $\lambda_k$ and therefore $P(\lambda_k = \lambda^u_0) > 0$ and $P(\lambda_k = \lambda^u_H - \delta) > 0$. Indeed, if $u \notin H_k$, $P(\lambda_k = \lambda^u_0) = P(\lambda_k = \lambda^u_H - \delta) = 0$. Since an appropriate strategy always selects $u \in H_k$ when $H_k \neq \emptyset$, the probabilities $P(\lambda_k = \lambda^u_0)$ and $P(\lambda_k = \lambda^u_H - \delta)$ observed at each $k$ by an appropriate strategy are at least as large as those observed by a non-appropriate strategy. As a consequence, the expected convergence time $E[T]$ of an appropriate strategy cannot be larger than $\sum_{u \in U} E[T^u]$ which, according to (25), is upper bounded by $\frac{3}{2} P_{\min}$. \hfill \Box

APPENDIX B
PROOF OF LEMMA 2

This lemma is proved by induction.

Step 1: First, we show that the inequality in (21) holds for $z_k \subseteq x_k$ such that $z^1_k = 0$ or $z^2_k = 0$. Let us consider $z^3_k = 0$ (all the steps are equivalent for $z^2_k = \emptyset$). First, we need to prove two preliminary inequalities:

1) Inequality 1:

$$E_\lambda [J^*(x^1_k, x^2_k, \lambda)] > E_\lambda [J^*(\emptyset, x^2_k, \lambda)]$$

(26)

if $x^1_k \neq \emptyset$. This inequality comes from the fact that reaching a state with $x^1_k = \emptyset$ from another state with $x^1_k \neq \emptyset$ takes at least 1 time-slot, therefore $E_\lambda [J^*(x^1_k, x^2_k, \lambda)] \geq 1 + E_\lambda [J^*(\emptyset, x^2_k, \lambda)] > E_\lambda [J^*(\emptyset, x^2_k, \lambda)],$

2) Inequality 2:

$$E_\lambda [J^*(\emptyset, x^2_k, \lambda)] > E_\lambda [J^*(\emptyset, z^3_k, \lambda)]$$

(27)

if $z^3_k \subseteq x^2_k$. For any $x^2$ we have that

$$E_\lambda [J^*(\emptyset, x^2, \lambda)] = P_1(x^2) \frac{1}{P(\lambda^u_0)} + P_2(x^2) \frac{1}{P(\lambda^u_H - \delta)} + P_3(x^2) E[T^2]$$

(28)

where

$$P_1(x^2) = P(\lambda^u_0 \in x^2, \lambda^u_H - \delta \notin x^2)$$

$$P_2(x^2) = P(\lambda^u_0 \notin x^2, \lambda^u_H - \delta \notin x^2)$$

$$P_3(x^2) = P(\lambda^u_0 \in x^2, \lambda^u_H - \delta \in x^2)$$

(29)

and $E[T^2]$ is given by (24). If $z^3_k \subseteq x^2$, then we have $\sum_{i=1}^{\min} P_i(x^2) > \sum_{i=1}^{\min} P_i(z^2)$, which results in

$$P_3(x^2) > P_3(z^2) + \sum_{i=1}^{\min} (P_i(z^2) - P_i(x^2))$$

(30)

We can now obtain (27) for $z^3_k \subseteq x^2$ as follows

$$E_\lambda [J^*(\emptyset, x^2, \lambda)]$$

$$= P_1(x^2) \frac{1}{P(\lambda^u_0)} + P_2(x^2) \frac{1}{P(\lambda^u_H - \delta)} + P_3(x^2) E[T^2]$$

$$> P_1(x^2) \frac{1}{P(\lambda^u_0)} + P_2(x^2) \frac{1}{P(\lambda^u_H - \delta)} + P_3(z^2) E[T^2]$$

$$+ (P_2(z^2) - P_2(x^2)) E[T^2]$$

$$+ (P_3(z^2) - P_3(x^2)) E[T^2]$$

$$> P_1(x^2) \frac{1}{P(\lambda^u_0)} + P_2(x^2) \frac{1}{P(\lambda^u_H - \delta)} + P_3(z^2) E[T^2]$$

$$+ P_2(z^2) - P_2(x^2) + P_3(z^2) - P_3(x^2)$$

$$= P_3(z^2) \frac{1}{P(\lambda^u_0)} + P_2(z^2) \frac{1}{P(\lambda^u_H - \delta)} + P_3(z^2) E[T^2]$$

$$= E_\lambda [J^*(\emptyset, z^3_k, \lambda)]$$

(31)

where the first inequality comes from (30), and the second inequality comes from the fact that $E[T^2] \geq \frac{1}{P(\lambda^u_0)}$ and $E[T^2] \geq \frac{1}{P(\lambda^u_H - \delta)}$. Bellman’s equation (19) for $z_k$ is

$$J^*(\emptyset, z^3_k, \lambda) = 1 + E[J^*(\emptyset, z^3_{k+1}, \lambda)|z_k, \lambda_k].$$

(32)
By inequality 1 we have that

$$E[J^*(x_{k+1}, z_k^l, \lambda)|x_k, \lambda_k] \geq E[J^*(\emptyset, x_k^l, \lambda)|x_k, \lambda_k]$$

and by inequality 2 we have that

$$E[J^*(\emptyset, z_{k+1}^l, \lambda)|x_k, \lambda_k] > E[J^*(\emptyset, z_k^l, \lambda)|x_k, \lambda_k]$$

(33)

and therefore

$$J^*(x_k, \lambda_k) \geq J^*(\emptyset, z_k^l, \lambda_k).$$

(4)

Step 2: The induction step consists of showing that if the inequality (21) holds for uncertainty region vectors contained in $x_k$, then it holds for $x_k$, i.e., we assume that given $z_k$ such that $z_k \subseteq x_k$, the inequality $J^*(x_{k+1}, \lambda_k) \geq J^*(y_k, \lambda_k)$ holds for any $y_k \subseteq z_k$, and then we show that this implies $E[J^*(x_{k+1}, z_k^l, \lambda)|x_k, \lambda_k]$.

Because $z_k \subseteq x_k$, we have $(z_{k+1}^l, z_k^l) \subseteq (x_{k+1}^l, x_k^l)$, $(z_{k+1}^l, z_k^l) \subseteq (x_{k+1}^l, x_k^l)$, and $(z_{k+1}^l, z_k^l) \subseteq (x_{k+1}^l, x_k^l)$, where the vectors in the right hand side of $\subseteq$ are contained in $x_k$ (thus equivalent to $z_k$ in the induction assumption), and the vectors in the left hand side of $\subseteq$ are contained in $z_k$ (thus equivalent to $y_k$ in the induction assumption). Therefore, by the induction assumption we have

$$E[J^*(z_{k+1}, z_{k+1}^l, \lambda)] \geq E[J^*(z_{k+1}, z_k^l, \lambda)]$$

$$E[J^*(z_{k+1}, z_{k+1}^l, \lambda)] \geq E[J^*(z_{k+1}, z_k^l, \lambda)]$$

$$E[J^*(z_{k+1}, z_{k+1}^l, \lambda)] \geq E[J^*(z_{k+1}, z_k^l, \lambda)]$$

(34)

and therefore $J^*(x_k, \lambda_k) \geq J^*(z_k^l, \lambda_k)$. □

REFERENCES


Jose A. Ayala-Romero received the B.Sc. degree in telecommunications and the M.Sc. degree in telecommunication engineering from the Technical University of Cartagena (UPCT), Spain, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Department of Information and Communications Technologies. He was a Visiting Ph.D. Student with the Department of Information Engineering, University of Padova, in 2017, and NEC Laboratories Europe, Germany, in 2018. His current research focuses on machine learning algorithms for wireless networks management.

Juan J. Alcaraz received the M.Sc. degree in telecommunications engineering from the Polytechnic University of Valencia, Spain, in 1999, and the Ph.D. degree in telecommunications engineering from the Technical University of Cartagena (UPCT), Spain, in 2007. From 1999 to 2004, he was with several technology companies until joining UPCT in 2004, where he is currently an Associate Professor with the Department of Information and Communication Technologies. He was a Fulbright Visiting Scholar with the Electrical Engineering Department, University of California at Los Angeles (UCLA), in 2013, and a Visiting Professor with the Department of Information Engineering, University of Padova, in 2017. His current research focuses on learning algorithms for wireless networks.

Andrea Zanella (S’98–M’01–SM’13) received the Laurea degree in computer engineering and the Ph.D. degree in electronic and telecommunications engineering from the University of Padova, Italy, in 1998 and 2001, respectively. In 2000, he was a Visiting Scholar with the Department of Computer Science, University of California at Los Angeles (UCLA). He is one of the coordinators of the SIGnals and NETworking (SIGNET) Research Lab. He is currently an Associate Professor with the Department of Information Engineering (DEI), University of Padova. His long-established research activities are in the fields of protocol design, optimization, and performance evaluation of wired and wireless networks. He is also a Technical Area Editor of the IEEE INTERNET OF THINGS JOURNAL and an Associate Editor of the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and Digital Communications and Networks (DCN).

Michele Zorzi (F’07) received the Laurea and Ph.D. degrees in electrical engineering from the University of Padova in 1990 and 1994, respectively. From 1992 to 1993, he was on leave at the University of California at San Diego (UCSD). In 1993, he joined the Faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications, UCSD, he joined the School of Engineering, University of Ferrara, Italy, in 1998, where he became a Professor in 2000. Since 2003, he has been with the Faculty of the Information Engineering Department, University of Padova. His current research interests include performance evaluation in mobile communications systems, WSN and Internet of Things, cognitive communications and networking, vehicular networks, 5G mm-wave cellular systems, and underwater communications and networks. He was a recipient of several awards from the IEEE Communications Society, including the Best Tutorial Paper Award in 2008, the Education Award in 2016, and the Stephen O. Rice Best Paper Award in 2018. He was the Editor-In-Chief of the IEEE WIRELESS COMMUNICATIONS from 2003 to 2005, the IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 to 2011, and the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING from 2014 to 2018. He has served as a Member-at-Large for the Board of Governors of the IEEE Communications Society from 2009 to 2011 and the Director of Education from 2014 to 2015.
Conclusions

Overview of the problem relevance

Heterogeneous networks are envisioned to be a key element in the development of 5G and beyond technologies. This architecture relies on the deployment of small cells within the macro cell coverage area and presents many advantages, such as frequency reuse, load balancing, and lower latency.

There are, however, some technical challenges still unsolved regarding this architecture. In this thesis, we address two of the most important ones: the inter-cell interference and the power consumption. We adopt the SON paradigm, from which the self-optimizing capability allows the network to autonomously configure its parameters in an online manner, that is, while the network is operating.

In order to implement the self-optimizing capability, we propose the use of online learning algorithms. This approach has many advantages with respect to previous ones: first, online learning algorithms do not need a predefined model of the network, which is seen as a black box. Second, they work directly with performance measurements from the network, not requiring the performance metric to have a closed analytical form. Third, they do not need network parameters as input (e.g., channel state, user location). These parameters can be unknown or unavailable and may increase the computational and signaling overhead. Fourth, they have, in general, good adaptability to changes in the network state.

Summary of Contributions

In this section, we summarize the contributions of this thesis, which can be divided into two parts.

In the first part, we addressed the interference coordination problem. In Chapter 3, we have proposed an online learning stochastic optimization algorithm known as Response Surface Methodology (RSM) capable of configuring the eICIC parameters. This algorithm takes performance samples from the operating network and, based on these samples, it computes a noisy estimation of the steepest-ascent direction, aiming at approaching the optimum. Once the algorithm has approached the optimum, it can track variations on the optimal configuration produced by changes in the network conditions. Our numerical results showed that the proposal presents a quick convergence and can effectively track the changes in the network conditions.
We extended this solution in Chapter 4 with a new proposal referred to as Stepwise-Restricted RSM (SR-RSM), which includes stochastic restrictions. Thus, we take into account the trade-offs among different metrics by optimizing one metric while keeping the others above their respective desired thresholds. The novelty of SR-RSM relies basically on two ideas. First, a new search direction, taking into account the constraints, which has been derived using interior point methods. This new search direction increases the convergence rate by avoiding hitting the restriction boundaries. Second, we perform as many hypothesis tests as constraints we have at each iteration, intending to assess if the constraints will be fulfilled in the next step. If the hypothesis tests are rejected (the next configuration meets the constraints), the algorithm continues to the next configuration; otherwise, the algorithm remains in the current configuration and repeats the iteration. Our numerical results showed a clear advantage in terms of convergence velocity and performance with respect to previous proposals.

In Chapter 5, we presented a new proposal referred to as DDeC, which is based on a Multi-Armed Bandit algorithm. It comprises three modules: The Action Availability Selector (AAS) selects at each time stage the set of available controls. The Multi-Armed Bandit with limited action availability (MAB-LAA) selects which control to use at each time step from the set of available controls generated by the AAS. Finally, the Network Change Detector (NCD) observes the performance samples from the network in order to detect variations in network conditions. When significant variations are detected, the NCD notifies the other modules.

These modules work together in two operation modes: search and exploitation. The objective of the search period is to approach the region of the optimal configuration as fast as possible. When the algorithm reaches the optimal region, the exploitation period starts. During this operation mode, the algorithm performs a fine-tuned exploration of the action set and avoids deviating from the optimal region. Thus, the exploration is reduced, which increases its performance. Another advantage of this proposal with respect to previous ones is that, as our numerical results have shown, its performance is not highly affected by the configuration of its parameters.

In the second part, we addressed the joint problem of energy-saving and interference coordination. In Chapter 6, we formulated the problem as a finite horizon MDP aiming at minimizing the overall power consumption in the network. We also consider one constraint in the formulation: a minimum QoS requirement has to be satisfied at every time stage. Since solving the MDP is computationally unaffordable, we proposed an approximate dynamic programming algorithm using the CEC principle. In order to forecast the power consumption and the QoS in future time stages, we use machine learning tools. This allowed us to estimate the cost-to-go function and apply the CEC principle.

In Chapters 7 and 8, we formulated the problem as a contextual MAB problem which implies several advantages: This formulation allows us to configure the duration of a stage (time between two consecutive configurations) to any value and even change this value depending on the velocity of the traffic load variation. It does not assume a predefined traffic pattern, which is especially useful under special
events such as concerts, sports games, etc. We proposed two algorithms addressing this problem: ClassMAB and Bayesian Response Estimation and Threshold Search (BRETS), in Chapters 7 and 8, respectively.

ClassMAB comprises two parts: a neural network (NN) classifier and a Multi-Armed Bandit (MAB) algorithm. The NN allows us to obtain the set of QoS feasible controls for a given network state, and the MAB learns the most efficient ones in terms of power consumption among them. This online learning proposal obtains excellent results in terms of regret after the learning process and achieves an energy saving of 19.48% with respect to a default configuration.

On the other hand, BRETS comprises two parallel learning processes. The first one, Threshold Search, is a new mechanism based on the premise that, for each control action, the QoS objective is fulfilled only when the UE traffic intensity is below some threshold. The traffic threshold for each control is initially unknown, and the efficient discovery of all the thresholds is its objective. The second learning process, Response Estimation, aims at estimating the function that maps traffic intensity to network energy consumption for each control. Response Estimation uses a Bayesian approach, in particular, a Gaussian Process, which generates response function estimations with relatively few samples and provides the uncertainty of the estimation at each traffic intensity. We provide bounds for the expected convergence time of the algorithm.

This proposal presents several advantages detailed as follows. The Threshold Search algorithm leverages that the QoS function follows a threshold structure. This allows the algorithm to minimize the QoS failures and to converge faster than any previous proposal. Moreover, the use of Gaussian Processes in the Response Estimation allows the algorithm to apply the principle of optimism in the face of uncertainty used by classic bandit algorithms in a novel way: each configuration is associated with a function instead of a scalar. This improves the performance of the algorithm in terms of power consumption with respect to previous proposals. In particular, BRETS achieves a 25% of energy saving with respect to a default configuration.

**Future Work**

We believe the contributions of this thesis will be helpful in the development of future Self-organizing heterogeneous networks. The implementation of small cells is becoming easier and cheaper since some companies are taking steps toward the implantation of eNBs using open source software and commercial off-the-shelf (COTS) hardware. This is the case of srsLTE [88], an open source project developed by Software Radio Systems [89] which implements a complete software defined radio (SDR) LTE UE, and a complete SDR LTE eNB with all layers from PHY to IP. srsLTE also implements a light-weight LTE core network (EPC) implementation with Mobility Management Entity (MME), Home Subscriber Server (HSS), and Serving and Packet Data Network Gateway (S/P-GW).

On the other hand, OpenAirInterface [90] is supported by a non-profit alliance of industrial and research contributors aiming at developing open source software
Conclusions

and hardware for the core network (EPC), access network, and user equipment (EUTRAN) of 3GPP mobile networks.

These open source implementations of the LTE standards have a high potential. Fujitsu recently showed it in an experiment in which they connected 40 COTS UEs to an OpenAirInterface LTE eNB and EPC, allowing them to consume video streaming [91]. This demonstrates that the current OpenAirInterface eNB software is ready and tested to be used in many environments such as private LTE networks and rural deployments.

This opens new research lines applying the ideas developed in this thesis in experimental testbeds. The open source implementation of a softwarized eNB allows an easy and low cost implementation of a virtualized radio access network (vRAN) architecture based on general-purpose processing platforms (GPPP) and decoupled base band units (BBUs) and remote radio units (RRUs). Although this architecture presents several advantages, the computing resource are generally over-dimensioned to satisfy peak demands [92]. Conversely, the dynamical allocation of computational resources can lead to significant savings in terms of CPU usage and power consumption.

The application of model-free online learning algorithms to this problem is promising since the relation between the load of the eNB, the state of the radio channel of the UEs and the computing requirements is unknown and depends on the specific implementation (computing platform, implementation of the virtualized eNB, etc) [93].


