Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

RESEARCH ARTICLE

Online Reinforcement Learning for Adaptive Interference Coordination†

Juan J. Alcaraz*[1] | Jose A. Ayala-Romero[2] | Javier Vales-Alonso[1] | Fernando Losilla-Lopez[1]

[1]Department of Information and Communication Technologies, Universidad Politécnica de Cartagena, Murcia, Spain
[2]School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland

Correspondence
*Corresponding author: Juan J. Alcaraz.
Email: juan.alcaraz@upct.es

Summary

Heterogeneous Networks (HetNets), in which small cells overlay macro cells, are a cost-effective approach to increasing the capacity of cellular networks. However, HetNets have raised new issues related to cell association and interference management. In particular, the optimal configuration of interference coordination (IC) parameters is a challenging task because it depends on multiple stochastic processes such as the locations of the users, the traffic demands, or the strength of the received signals. This work proposes a self-optimization algorithm capable of finding the optimal configuration in an operating network. We address the problem using a Reinforcement Learning (RL) approach, in which the actions are the IC configurations, whose performances are initially unknown. The main difficulty is that, due to the variable network conditions, the performance of each action may change over time. Our proposal is based on two main elements: the sequential exploration of subsets of actions (exploration regions), and an optimal stopping strategy for deciding when to end current exploration and start a new one. For our algorithm, referred to as Local Exploration with Optimal Stopping (LEOS), we provide theoretical bounds on its long-term regret per sample and its convergence time. We compare LEOS to state-of-the-art learning algorithms based on multi-armed bandits and policy gradient RL. Considering different changing rates in the network conditions, our numerical results show that LEOS outperforms the first alternative by 22%, and the second one by 48% in terms of average regret per sample.

KEYWORDS:
Interference coordination, heterogeneous networks, reinforcement learning, online learning

1 | INTRODUCTION

The Heterogeneous Network (HetNet) architecture, in which multiple small cells overlay each macro cell, is used in current and forthcoming mobile access networks to increase spacial spectral reuse, enhance network coverage, and reduce the load of the macro cells (offloading)[1]. To increase spectrum efficiency, HetNets need to incorporate an interference management scheme, such as the enhanced Inter-Cell Interference Coordination (eICIC) mechanism defined by the Third Generation Partnership Project (3GPP) for Long Term Evolution Advanced (LTE-A) networks[2]. This mechanism is expected to remain relevant for 5G deployments[3], because of their higher density and the incorporation of LTE in the 5G radio access thanks to the network slicing

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

functionality. Additionally, there are several interference management proposals for 5G New Radio (5G NR) similar to eICIC[4]. In eICIC, 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 parameters: *Cell Range Extension* (CRE) bias, and *Almost Blank Subframe* (ABS) ratio.

The CRE bias is an offset added to the small cell power level received at the user terminals, favouring the user association to the small cell instead of the macro cell, thus determining the effective range of the small cell. The ABS ratio refers to the proportion of radio access subframes that are muted by the macro cell in order to alleviate the interference experienced by the small cell users. The configuration of these parameters is not standardized, and is a challenging issue because they have a joint effect on the user throughput, and must be continuously readjusted due to the random changes in the network conditions[5].

Earlier works on eICIC[6,7,8,9,10] relied, in general, on mathematical modeling. This approach is appropriate, e.g., to estimate the network capacity, but not to optimize the configuration of a real operating network, because of the limitations of the mathematical models. For example, there are no accurate models allowing mathematical tractability of the 5$^{\text{th}}$ percentile user throughput, which is defined by the 3GPP[11] to characterize the performance of LTE networks.

In contrast, we follow a data-based approach. This means that we do not try to *predict* the system performance using a mathematical model characterizing propagation effects, traffic behaviors, user movement patterns, and so forth. Instead, we propose a scheme that applies configurations to the system, and *observes* the resulting performance. These observations allow a controller to learn the best configuration. Of course this must be done in a cautionary and efficient way. A naive strategy would simply sample every possible configuration repeatedly, estimate their performances statistically, and then select the empirical best one. However, this extensive and indiscriminate sampling scheme is harmful to the user performance and unsuitable for online operation under time-varying conditions.

The reduction of the performance loss due to non-optimal samples is the main objective of Multi-Armed Bandit (MAB) algorithms[12]. Besides, the system response to eICIC configurations has been shown to be *unimodal*[13], which allows the MAB algorithms to be sequentially applied in small subsets of actions (Local Exploration, LE), following an iterative process that resembles a stochastic gradient ascent[5]. Exploiting unimodality boosts the performance of MAB algorithms. However, the main difficulty for applying this strategy in our scenario is that the network conditions such as the traffic intensity, spatial distribution of the UEs and others, change over time, implying that the expected performance at each action is time-varying. This variability raises challenging questions such as: what is the optimal duration of each LE stage? Can we track the optimal configuration as it changes? At what cost?

To solve the above issues, we propose an algorithm that combines Reinforcement Learning (RL) and MAB for unimodal responses. The main idea is to dynamically adjust the time spent on each LE stage by making an optimal stopping (OS) decision after every observation. OS problems can be typically addressed by RL, but in our case it is a challenging task because the change rate of the system is unknown, and because of the large dimension of the state space, which comprises all the possible histories of past observations (selected actions and observed performances). Our solution strategy circumvents these difficulties by leveraging the structure of the problem, casting it into a sequential likelihood ratio test for which our algorithm learns how to generate an effective decision boundary. The main novelty of our proposal relies on exploiting this latter idea to design a model-based RL approach for solving the OS problem.

Figure 1 summarizes our proposal: A MAB algorithm selects the actions within each local exploration stage, while a RL-based OS policy decides whether to continue or to end current local exploration. The parameters of the OS policy can be learned offline using network data. Therefore, our proposal combines strategies from both MAB and RL algorithms.

The rest of the paper is organized as follows. Section 2 overviews related works and highlights our contributions. The main problem is discussed in Section 3. Our proposal is presented in Section 4 and Section 5 describes two relevant alternatives: an existing algorithm for structured non-stationary MABs, and a reinforcement learning approach based on policy gradient. Section 6 provides performance bounds for our proposal. In Section 7 we numerically evaluate and compare our proposal with the alternatives and finally, Section 8 summarizes the main conclusions of this work and outlines future research lines.

2 | RELATED WORK AND CONTRIBUTION

2.1 | Related Work

There is an emerging interest in the use of data-based techniques in network management and optimization[24,25,26]. In particular[24] proposes a framework for data-driven network optimization and points out several potential applications of this framework

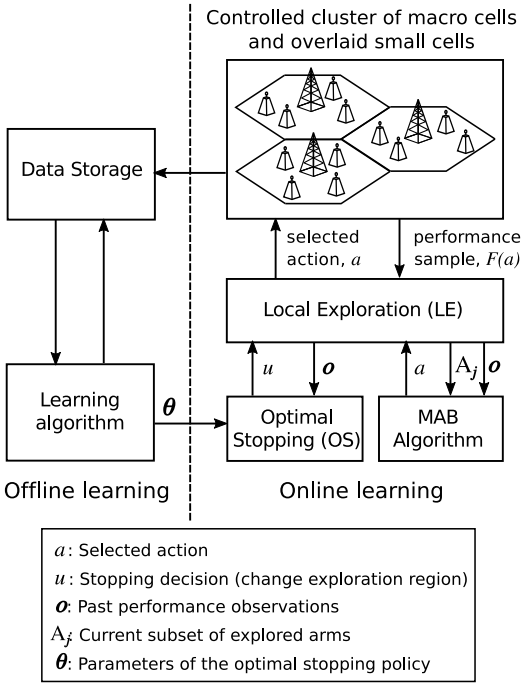

FIGURE 1 Block diagram of our proposal Local Exploration with Optimal Stopping (LEOS).

	14,15	16	17	18,19	20,21	22	23	**Ours**
Controlled Parameters	CRE	CRE	ABS	ABS	ABS	ABS+CRE	ABS+CRE	**ABS+CRE**
Non-stationary scenario	No	No	No	Yes	Yes	Yes	Yes	**Yes**
Approach	Heuristic	RL	RL	Heuristic	Optimization	Heuristic	RL	**RL**
Model free	No	No	No	No	No	No	No	**Yes**

TABLE 1 Comparison of related works

including interference management. Our paper develops this idea and proposes a specific novel algorithm applicable to eICIC configuration.

Previous works have addressed the efficient configuration of eICIC parameters. Some works focused solely on the CRE bias [14,15,16], others focused on the ABS ratio [17,18,19,20,21,27], while fewer works [22,23,28] aimed at optimizing both parameters simultaneously, like this work.

Regarding the dynamic nature of the network conditions (e.g. traffic intensity, number of users, user positions), most previous works [7,16,29,8,30,17,14,15,31,20,32,21,33,34] assume static or stationary conditions. While some works consider non-stationary scenarios [18,19,22,9,28], their approaches are based on optimization or heuristic algorithms over *mathematical models of the network*, like in [6,7,8,10,35,36,20,32,21,34,37]. For mathematical tractability, even the most complete mathematical network models comprise simplifications and assumptions that may limit their application to real operating networks. For example, the authors in [8] simplify their model by considering only the strongest interference at the receivers. In contrast, our proposal learns directly from the data gathered from the network and does not require any previous knowledge or assumption, our approach is essentially *model-free*. A usual modeling tool is stochastic geometry [35,36], which assumes that nodes are deployed according to spatial distributions like Poisson point processes. Another usual simplification consists of assuming round robin scheduling on the radio interface [35], which is not well suited for HetNet environments.

Reinforcement learning (RL) algorithms have been previously considered for this task, mainly under stationary network conditions [16,17] but also considering non stationary traffic conditions [23,37] as in our approach. 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 order to tackle with the dimensionality and the non-stationarity of the system, our RL approach adopts the two key techniques

discussed in the introduction: local exploration and sliding window. Our previous work[5] was the first one to use a RL approach for self-optimization of interference management parameters, exploiting the unimodality of the system response. However, that approach was not designed for a continuous change of the system response which is the central aspect of this new work. Table 1 summarizes the main aspects of the mentioned works with respect to ours.

2.2 | Contribution

This paper proposes a novel self-optimization algorithm for interference management in HetNets allowing a centralized entity to find and track efficient eICIC configurations in an operating LTE-A network, adapting to variable network conditions and leveraging past experience. The main features of our proposal are:

- We propose a novel algorithm for non-stationary MAB problems based on local exploration combined with an optimal stopping policy that controls the duration of each local exploration stage.
- We propose an effective strategy for approximating the optimal stopping policy based on some structural properties of the problem allowing us to cast it into a sequential likelihood ratio test.
- We provide theoretical performance bounds in terms of regret and convergence time.
- We show that, compared to other state-of-the-art alternatives, our proposal performs better under changing network conditions.

Note that the proposed algorithm, LEOS, can be applied not only for automatic eICIC configuration, but for the self-optimization of any mechanism whose optimal parameter setting varies over time in a non-stationary way, provided the number of parameters is sufficiently small, as for eICIC.

3 | THE INTERFERENCE COORDINATION PROBLEM

3.1 | eICIC Parameters and Performance Metric

The eICIC functionality was introduced by the 3GPP in Release 10 (LTE-A)[2] 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 $(0/8, \dots, 8/8)$ within the ABS pattern. As in[38,35], and following the 3GPP recommendation[11], we consider that the controlled cluster of cells share the same ABS pattern (synchronized muting) and the same CRE bias. The cells in the cluster are assumed to have similar traffic patterns, e.g. the cluster covers a residential area of similar population density.

To evaluate the performance of LTE networks, the 3GPP proposes the 5^{th} percentile throughput[39], which is also used in several previous works on interference coordination[6,18,38]. We define a *throughput sample* as the quotient between the size of a downloaded file and the time required to download it. The 5^{th} percentile throughput is defined as the value below which 5% of the throughput samples fall. In other words, 95% of the throughput samples are above the 5^{th} percentile value. In a practical setting, we obtain estimations of this metric from a finite set of throughput samples coming from the users within the controlled cluster of cells. Each estimation is then a random variable that we refer to as *performance sample* or performance observation. Note that this metric characterizes the worst performing UEs, and thus maximizing its expectation pursues max-min fairness among the UEs in the network.

3.2 | Problem Formulation

Let $\gamma \in \Gamma$ and $\phi \in \Phi$ refer to the ABS ratio and CRE bias respectively, with $\Gamma = \{\gamma_1, \dots \gamma_{M_1}\}$ and $\Phi = \{\phi_1, \dots \phi_{M_2}\}$, such that $\gamma_1 < \dots < \gamma_{M_1}$ and $\phi_1 < \dots < \phi_{M_2}$. Any configuration pair $a = (\gamma, \phi)$ must be selected from the set $\mathcal{A} = \Gamma \times \Phi$ containing $M = M_1 M_2$ elements. Time is divided into time-slots, $n = 1, 2, \dots$. The decision maker selects a configuration $a \in \mathcal{A}$ at the beginning of each time-slot, and obtains the corresponding performance sample $F_n(a)$ at the end of the time-slot, which is an estimation of the 5$^{\text{th}}$ percentile throughput as defined in previous subsection. For each a, the observation $F_n(a)$ is a random variable whose distribution is initially unknown. We refer to the set of M distributions associated to the M actions, as the response of the system ($\mathcal{F}_n$). Note that this response changes over time. In particular, it is assumed that $\mathcal{F}_n$ remains unchanged during an unknown number of time-slots, until some event causes $\mathcal{F}_n$ to change (e.g. a variation in the number of UEs under coverage, a small cell being switched off, etc)

Our goal is to devise a strategy or *policy* that determines which configuration a should be selected at each time-slot, considering the past history of selected configurations and performance samples, with the objective of maximizing the expected performance over time.

Let us first consider only periods in which the response remains unchanged. In this case, the problem can be formulated as a stochastic multi-armed bandit problem, characterized by M actions (configuration pairs). Let $a^* = \arg\max_{a \in \mathcal{A}} E[F_n(a)]$ be the best performing action, and let $\mu^* = E[F_n(a^*)]$. Consider a policy η that selects an action $\eta(n) \in \mathcal{A}$ at each time-slot n. We define the *instantaneous regret* r_n as the difference between the expected reward (performance) of a^*, and the performance of the selected action $\eta(n)$.

$$r_n = \mu^* - E[F_n(\eta(n))]. \tag{1}$$

And we define the *accumulated regret* or simply regret $R(n)$ as the difference between the total expected reward obtained by always selecting a^*, and the expected total reward obtained by η up to time n

$$R(n) = E\left[\sum_{n'=1}^{n} r_{n'}\right] = n\mu^* - E\left[\sum_{n'=1}^{n} F(\eta(n'))\right]. \tag{2}$$

The objective of a bandit algorithm is, in general, to find a policy η minimizing $R(n)$. There is an extensive literature addressing this problem.

However, conventional bandit algorithms are not applicable in our case because of the variability of $\mathcal{F}_n$ over time. This variability implies that the optimal action a^* also changes over time. Our algorithm should be able to adapt to these changes during the system operation, that is, it must find *and track* a^* during future changes in the system response. Note that we assume that $\mathcal{F}_n$ remains unchanged long enough for an *efficient* learning algorithm to find a^* for this specific system response.

This assumption highlights the need to find a^* within as few samples as possible. To this aim, we leverage the underlying structure of the expected rewards μ_a for $a \in \mathcal{A}$ for a given $\mathcal{F}_n$. Next subsection describes this structure.

To facilitate the reading, Table 2 summarizes the main notation used in the paper.

3.3 | Unimodal Structure

In this subsection we describe how the 5$^{\text{th}}$ percentile throughput is affected by variations on the ABS ratio and CRE bias, resulting in the unimodal structure of the set of expected rewards μ_a for $a \in \mathcal{A}$. Let us refer to the users in the CRE regions as CRE UEs, and the users in the macro cells (not covered by any small cell) as macro UEs. In general, in absence of ABS, the smallest throughput samples come from CRE UEs because of their poor SINR. For a given CRE bias, increasing the ABS ratio implies two things: less interference at the CRE UEs and fewer spectral resources for the macro UEs. In consequence, using a larger ABS ratio increases the 5$^{\text{th}}$ percentile throughput as long as it is determined by the CRE UEs, who are increasing their throughput. However, there exists some point at which the 5$^{\text{th}}$ percentile throughput starts to be determined by the macro UEs, who receive fewer frame resources as the ABS ratio increases. Therefore, beyond this ratio, the 5$^{\text{th}}$ percentile throughput starts to decay, because of the reduced throughput of some macro users.

A similar reasoning can be made when considering a given ABS ratio. As the CRE bias increases, more macro UEs become CRE UEs. Therefore, using a larger CRE bias increases the 5$^{\text{th}}$ percentile throughput as long as this implies that there are more available frame resources per macro UE and per CRE UE (i.e. as the resources become better balanced). However, increasing the CRE bias beyond the value that attains the optimal resource allocation can only deteriorate the performance since increasing the number of CRE UEs will always imply fewer resources for them.

Notation	Description
$\mathcal{A}$	Set of actions (eICIC configuration pairs) available
M	Total number of actions ($\|\mathcal{A}\|$)
$F_n(a)$	Random performance observation at $a \in \mathcal{A}$
μ_a	Expected performance of action a: $\mu_a = E[F_n(a)]$ at a given stage n
σ	Standard deviation of $F_n(a)$ at a given stage n
$\hat{\mu}_a$	Estimated performance of action a
$\hat{\psi}_a$	Upper bound estimate of the action a performance
$\mathcal{S}$	Exploration regions of LEOS
$\mathcal{A}_j$	Set of actions in exploration region j
m	Number of actions of each exploration region ($\|\mathcal{A}_j\|$)
a_j^*, $\hat{a}_j$, a_j	Best local action, estimated best local action, and central action of region j
$\mathbf{o}_n$	Past observations (action, performance) at time-slot n
$\{Y_k\}$	DTMC modeling the exploration regions visited by LESH
$p_{j,j'}$	Transition probabilities for $\{Y_k\}$
$\{X_k\}$	Worst case DTMC for obtaining performance bounds on $\{Y_k\}$
$\overline{\pi}$, π	Steady state probability vectors for $\{Y_k\}$ and $\{X_k\}$ respectively
Δ_a	Average performance gap between action a and the best performing action, a^*
r_n	Instantaneous regret of the n-th sample
$R(n)$	Expected accumulated regret up to sample n
N	Sample budget for each local exploration
$E[t]$	Expected length of an exploration stage
$E[t_j]$	Expected length of an exploration stage in $\mathcal{A}_j$
p_e	Upper bound for the estimation error probability

TABLE 2 Notation

In order to illustrate the described behavior, Figure 2 shows the system responses, in terms of 5$^{\text{th}}$ percentile throughput, of a LTE-A network under different conditions (traffic intensity and number of active small cells). The network was simulated according to the 3GPP guidelines (described in Section 7). The described effects were previously reported and exploited in other works like [35] and [13].

To formalize the above property in our model, we can construct an undirected graph $\mathcal{G}$, using $\mathcal{A}$ as the set of vertices, and defining its set of edges $\mathcal{E}$ as follows: For each action $a = (\gamma_i, \phi_{i'}) \in \mathcal{A}$, the set $\mathcal{E}$ contains up to 8 edges[1] pairing a with each action (γ, ϕ) such that $\gamma \in \{\gamma_{i-1}, \gamma_i, \gamma_{i+1}\}$, and $\phi \in \{\phi_{i'-1}, \phi_{i'}, \phi_{i'+1}\}$. We say that two vertices (actions) a, a' are neighbors if $(a, a') \in \mathcal{E}$. The distance between two actions $d(a, a')$ is defined as the number of edges of the shortest path between a and a' in $\mathcal{G}$.

Unimodality in graphs expresses the fact that when the optimal action is a^*, then for *all* $a \in \mathcal{A}$, there exists a path in $\mathcal{G}$ from a to a^* along which the expected reward is strictly increasing. The consequence of this definition is that there are no local maxima except at a^*. We have defined $\mathcal{G}$ for two parameters γ and ϕ (two-dimensional case). If we only consider the configuration of one parameter, say γ, then $\mathcal{A} = \Gamma$, and unimodality implies that there exists some $\gamma_{i^*} \in \mathcal{A}$ such that $\mu_1 < \ldots < \mu_{i^*-1} < \mu_{i^*}$ and $\mu_{i^*} > \mu_{i^*+1} \ldots > \mu_{M_1}$, where μ_i denotes the expected reward of γ_i (one-dimensional case).

4 | LOCAL EXPLORATION WITH OPTIMAL STOPING

In this section we describe our proposal, Local Exploration with Optimal Stoping (LEOS), in detail. The main idea of LEOS is to obtain samples from a subset of neighboring actions (Local Exploration, LE), until the obtained observations provide sufficient evidence that moving to another subset would provide a better expected performance.

[1]The number of edges for a given a is fewer than 8 if a contains any of the minimum or maximum values γ or ϕ.

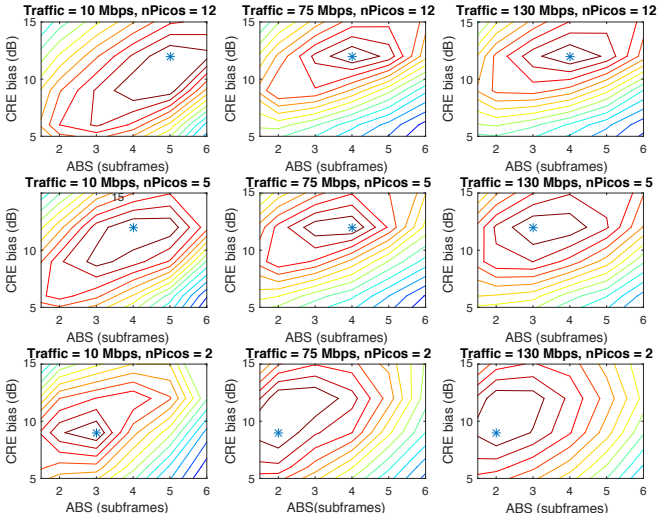

FIGURE 2 Contour line representations of the system response of a simulated LTE-A network under different conditions (traffic intensity and number of active small cells). The optimal configurations are identified with a $*$ symbol. In all cases we observe unimodal structure. The performance associated to each configuration is the estimated 5^{th} percentile throughput.

4.1 | Local Exploration

The Local Exploration algorithm requires the definition of S subsets of $\mathcal{A}$, referred to as *local exploration regions* or simply exploration regions. Each exploration region $\mathcal{A}_j$ for $j \in \mathcal{S} = \{1, \dots, S\}$, contains m actions, including a *central action* a_j, and its $m-1$ neighbors in $\mathcal{G}$. Given one of these neighbors, $a_{j'}$, we say that its corresponding region $\mathcal{A}_{j'}$ is adjacent to $\mathcal{A}_j$. We will consider exploration regions comprising 3 values of each configurable parameters, thus $m = 3 \times 3$.

We define, the *best local action* a_j^* of $\mathcal{A}_j$, as the action whose expected reward is the highest among the m actions in $\mathcal{A}_j$, and the *exploration stage*, as a sequence of up to N consecutive performance samples from a given exploration region $\mathcal{A}_j$. The objective of an exploration stage is to find a_j^*. We refer to N as the *sample budget* for each stage.

The LE algorithm operates in combination with other two algorithms: 1) A bandit-type algorithm (generically referred to as *SelectAction*), for deciding which action to select from $\mathcal{A}_j$, and 2) the Optimal Stopping (OS) algorithm, explained later in this section, for deciding when to finish current exploration stage. Both *SelectAction* and OS make their respective decisions according to the recent history of actions and observations. The LE algorithm operates as follows:

1. At every time-slot, *SelectAction* determines which action $a \in \mathcal{A}_j$ should be selected. The LE algorithm obtains a performance observation $F_n(a)$, and updates the observation history with pair $(a, F_n(a))$.
2. Given the observation history, the OS algorithm decides if current exploration stage should continue or not.
3. If current exploration stage must end, then the LE algorithm makes a *transition* to the adjacent region $\mathcal{A}_{j'}$ whose central action $a_{j'}$ is equal to the estimated best local action $\hat{a}_j$, and starts a new exploration stage in $\mathcal{A}_{j'}$.

Note that the self-transition $\mathcal{A}_j \rightarrow \mathcal{A}_j$ is possible. It corresponds to the case that the central action of A_j is identified as the optimal action. Figure 3 illustrates one transition of the LE algorithm in the two-dimensional case.

4.2 | Action Selection Algorithm

We use an upper confidence bound (UCB) strategy in *SelectAction*. At each exploration region $\mathcal{A}_j$, UCB needs to estimate, for each $a \in \mathcal{A}_j$, the average action performance $\hat{\mu}_a$, and an upper bound on this performance $\hat{\psi}_a$, at some fixed confidence level. In particular, we will consider the bounds based on Hoeffding's inequality [40] approximating the distribution of $F_n(a)$ by a Gaussian distribution with standard deviation σ (which can be a worst-case estimation based on past experience). The resulting expression for the upper bound estimate is given by $\hat{\psi}_a = \sqrt{\frac{6\sigma^2 \log t}{n_a}}$, where n_a denotes the number of samples taken from action a, and t is the total number of samples taken from $\mathcal{A}_j$. The action selected at each n is given by $\eta(n) \in \arg\max_{a \in \mathcal{A}_j} (\hat{\mu}_a + \hat{\psi}_a)$.

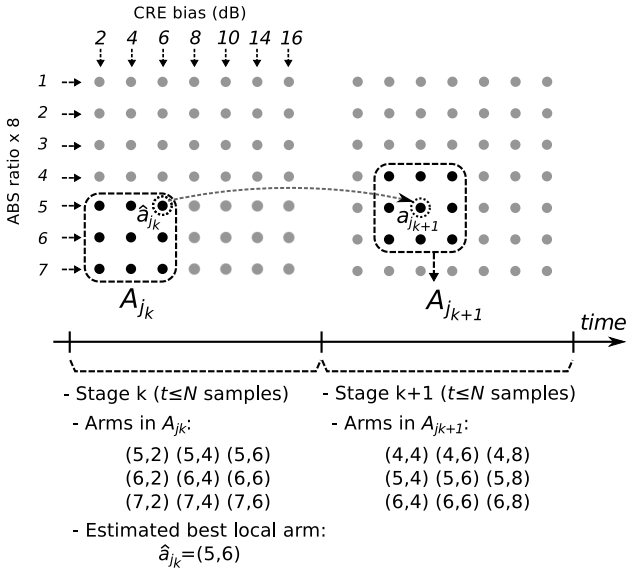

FIGURE 3 Operation example of the Local Exploration algorithm for ABS ratio and CRE bias configurations (two-dimensional case), using an exploration region of $m = 9$ actions.

To cope with the variability in the system response $\mathcal{F}_n$ over time, we will adopt a usual strategy in bandit algorithms consisting of limiting the amount of past observations upon which the algorithm bases its decisions[41,42]. In our case, the observation history (containing action-observation pairs) is limited to the past N time-slots:

$$\mathbf{o}_n = \left\{(\eta(n-N), F_n(\eta(n-N))), \ldots, (\eta(n), F_n(\eta(n)))\right\} \tag{3}$$

That is, $\hat{\mu}_a$ and $\hat{\psi}_a$ estimates are based on $\mathbf{o}_n$, which is updated following a *sliding window* procedure so that $\mathbf{o}_n$ always contains the N most recent action-observation pairs. Note that, even though $\mathbf{o}_n$ may contain samples from any $a \in \mathcal{A}$, only the samples from current exploration region are relevant.

4.3 | Optimal Stopping

Optimal stopping problems are a subclass of Markov decision processes (MDPs) whose objective is to decide, at each stage, if an ongoing process should continue or terminate. In our case, the ongoing process is the current local exploration, the stages of the MDP are the arrival times of the action-sample pairs, $n = 1, 2, \ldots$, and the state of the MDP is defined by the central action a_j of the current exploration region $\mathcal{A}_j$, and $\mathbf{o}_n$. Note that the dimension of the state increases with each action-sample pair, which makes the problem especially challenging. At each n, the decision maker must select between two actions: $u = 0$ to continue sampling from $\mathcal{A}_j$, or $u = 1$ to end current exploration stage. When the *stopping decision* $u = 1$ is made, the current local exploration ends, which implies that the MDP makes a transition to a *termination state* determined by the estimated best local action $\hat{a}_j$. Immediately after the termination of an MDP, a new exploration stage starts, which initiates a new MDP.

Let us define the optimal cost-to-go associated to a given state a_j at time-slot n, as the expected accumulated regret when starting at the exploration region whose central action is a_j, until reaching the optimal action a^* for the first time:

$$J_n(a_j) = \min_{\eta \in \Phi} E_{\mathcal{F},\eta}\left[\sum_{n'=n+1}^{n+T} r_{n'} \middle| a_j, \mathbf{o}_n\right] \tag{4}$$

where Φ denotes the set of possible policies (within the LE framework[2]), and T is a random variable defined as the number of samples required to reach a^* for the first time (convergence time). The subscripts in the expectation operator denote that the expectation is taken with respect to the stochastic process resulting by the policy η operating on a system response $\mathcal{F}$.

[2]Note that LE (in particular the action selection algorithm) specifies which action should be selected during the ongoing exploration stage, and LE also determines the next exploration region when a transition must be made. Thus, Φ contains the possible policies that the OS decisions can generate within the LE operation framework.

The optimal stopping decision should compare the cost-to-go when remaining at current state a_j, $J_n(a_j)$, with the termination cost, which is equal to the cost-to-go of a new sampling process starting at the estimated best local action $\hat{a}_j$, $J_n(\hat{a}_j)$. This can be expressed as

$$u = \arg\min_{u\in\{0,1\}} \Big[(1-u)J_n(a_j) + uJ_n(\hat{a}_j)\Big] \tag{5}$$

Let $e_n = P(\hat{a}_j \neq a_j^*|\mathbf{o}_n)$ denote probability that current estimation $\hat{a}_j$ is not the best local action a_j^*, i.e. e_n is the estimation error probability. Our solution method requires expressing the right hand side of (5) as a function of e_n. Note that, if $\hat{a}_j = a_j^*$, then $J_n(\hat{a}_j) = J_n(a_j^*)$. If $\hat{a}_j \neq a_j^*$, the expected cost-to-go at $\hat{a}_j$ is given by $E[J_n(\hat{a}_j)|\hat{a}_j \neq a_j^*, \mathbf{o}_n]$. Given these definitions, the optimal stopping decision is the one that solves, at each n, the following equation

$$u = \arg\min_{u\in\{0,1\}} \Big[(1-u)J_n(a_j) + u\big((1-e_n)J_n(a_j^*) + e_nE[J_n(\hat{a}_j)|\hat{a}_j \neq a_j^*, \mathbf{o}_n]\big)\Big] \tag{6}$$

If all the terms in (6) were known, finding an optimal stopping policy would be immediate. Unfortunately, this is not the case, and it is necessary to resort to approximate methods.

Our approach exploits the threshold structure of the optimal policy with respect to e_n. By rearranging the terms in (6), we obtain

$$u = \arg\min_{u\in\{0,1\}} \Big[J_n(a_j) + u\big[e_n\big(E[J_n(\hat{a}_j)|\hat{a}_j \neq a_j^*, \mathbf{o}_n] - J_n(a_j^*)\big) - (J_n(a_j) - J_n(a_j^*))\big]\Big] \tag{7}$$

from where it follows that the optimal decision is to select $u = 1$ when

$$e_n\big(E[J_n(\hat{a}_j)|\hat{a}_j \neq a_j^*, \mathbf{o}_n] - J_n(a_j^*)\big) < J_n(a_j) - J_n(a_j^*) \tag{8}$$

and $u = 0$ otherwise. Note that the inequality in (8) compares the expected performance loss of moving to a non optimal state $\hat{a}_j$, with the performance loss of remaining at a_j. If the former is smaller than the latter, then it is optimal to update the local exploration region (i.e. to make a stopping decision).

Defining

$$\gamma_n = \frac{J_n(a_j) - J_n(a_j^*)}{E[J_n(\hat{a}_j)|\hat{a}_j \neq a_j^*, \mathbf{o}_n] - J_n(a_j^*)}, \tag{9}$$

the optimal policy for (6) can be expressed as

$$\begin{aligned} &u = 1, \text{ if } e_n < \gamma_n \\ &u = 0, \text{ otherwise} \end{aligned} \tag{10}$$

In words, if the estimation error is below certain threshold, then a stopping decision should be made. Therefore, although e_n and γ_n are both unknown quantities, (10) indicates that an efficient strategy should be aware of how accurate the estimation $\hat{a}_j$ is, and use past observations to adjust the accuracy level required to make a transition.

4.4 | Sequential Likelihood Ratio Test

If γ_n were known, the optimal stopping decision would consist of assessing whether the estimation error probability e_n is below the threshold γ_n (10). This assessment can be done indirectly by means of a likelihood ratio test (LRT), as we explain in this subsection. In our case, the LRT aims at rejecting (or accepting) the null hypothesis $H_0 \equiv \hat{a}_j \neq a_j^*$[43]. The idea is to use $\mathbf{o}_n$ to estimate the likelihood ratio Z_n for H_0. The likelihood ratio is defined as the quotient between the probability of observing $\mathbf{o}_n$ conditioned on H_0 not being true, and the probability of observing $\mathbf{o}_n$ conditioned of H_0 being true. If Z_n surpasses a critical value $\xi(\gamma_n)$, then H_0 is rejected. By definition, the false rejection probability of the LRT is the estimation error probability, i.e. $P(Z_n > \xi(\gamma_n)|H_0) = e_n$. The larger the critical value $\xi(\gamma_n)$, the smaller the false rejection probability. Therefore the threshold condition in (10) could be replaced by $Z_n > \xi(\gamma_n)$. The diagram in Figure 4 illustrates the overall decision procedure presented in this section.

The ratio Z_n can be computed with the method explained in[43], consisting of two steps: First, we obtain $m-1$ pairwise likelihood ratios $Z_n(\hat{a}_j, a)$ for $a \in \mathcal{A}_j \setminus \{\hat{a}_j\}$, each of one aimed at rejecting (or accepting) the hypothesis "$\mu_{\hat{a}_j} < \mu_a$". The expression is given by

$$Z_n(\hat{a}_j, a) = n_{\hat{a}_j} D_{\mathrm{KL}}\left(\hat{\mu}_{\hat{a}_j}, \frac{n_{\hat{a}_j}\hat{\mu}_{\hat{a}_j} + n_a\hat{\mu}_a}{n_{\hat{a}_j} + n_a}\right) + n_a D_{\mathrm{KL}}\left(\hat{\mu}_a, \frac{n_{\hat{a}_j}\hat{\mu}_{\hat{a}_j} + n_a\hat{\mu}_a}{n_{\hat{a}_j} + n_a}\right) \tag{11}$$

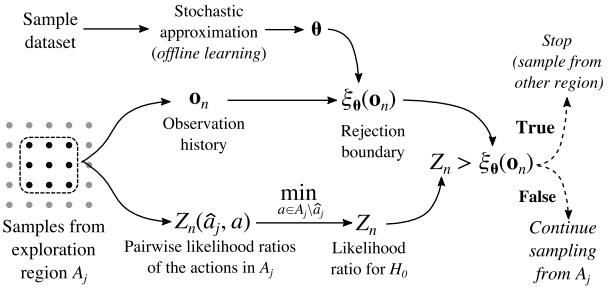

FIGURE 4 Diagram of the decision procedure used by LEOS with $Z_n > \xi_\theta(\mathbf{o}_n)$ approximating the optimal threshold condition $e_n < \gamma_n$.

where $D_{\mathrm{KL}}(\mu, \mu')$ denotes the Kullback-Liebler divergence between two distributions of the same type, having means μ and μ' respectively. Second, Z_n is given by

$$Z_n = \min_{a \in \mathcal{A}_j \setminus \{\hat{a}_j\}} Z_n(\hat{a}_j, a) \tag{12}$$

In summary, if the LTR rejects "$\mu_{\hat{a}_j} < \mu_a$" for all actions a in $\mathcal{A}_j$ different from $\hat{a}_j$, this means that it cannot assure that any of these actions has better performance than $\hat{a}_j$. As a consequence, the null hypothesis H_0 ($\hat{a}_j \neq a_j^*$) is rejected, implying that there is sufficient confidence on $\hat{a}_j$ being the best local arm.

Now we only need to tackle with one unknown parameter $\xi(\gamma_n)$, which depends on γ_n and thus on the unknown cost-to-go function J_n. Even if J_n could be estimated from the given observations $\mathbf{o}_n$ (e.g. by means of a reinforcement learning algorithm), the exact expression for ξ is unknown. Therefore, we propose to approximate ξ by an estimator $\xi_\theta(\mathbf{o}_n)$, characterized by a parameter vector $\boldsymbol{\theta}$. This vector can be fitted by means of a stochastic approximation algorithm. The objective is to find values of $\boldsymbol{\theta}$ that can approximately solve (4) using the following parametrized policy:

$$\begin{aligned} &u = 1, \text{ if } Z_n > \xi_{\boldsymbol{\theta}}(\mathbf{o}_n) \\ &u = 0, \text{ otherwise} \end{aligned} \tag{13}$$

In other words, our scheme consists of learning the best possible H_0 rejection boundary. Next section discusses our offline learning strategy for obtaining $\boldsymbol{\theta}$. The overall online learning proposal, LEOS, is summarized in Algorithm 1.

4.5 | Rejection Boundary Estimation

Let η_θ denote the action selection policy when the OS policy in (13) is used, and let $J_0(a_{j_0}; \boldsymbol{\theta})$ denote the cost-to-go from an initial state a_{j_0} associated to η_θ: $J_0(a_{j_0}; \boldsymbol{\theta}) = E_{F,\eta_\theta}\left[\sum_{n'=0}^{T} r_{n'} | a_{j_0}\right]$. By taking the expected value of $J_0(a_{j_0}; \boldsymbol{\theta})$ over the distribution of initial states $a_{j_0} \in \mathcal{A}$, we can define $J_0(\boldsymbol{\theta}) = E\left[J_0(a_{j_0}; \boldsymbol{\theta})\right]$. The estimation of the optimal rejection boundary is equivalent to solving the stochastic optimization problem $\min_{\boldsymbol{\theta}} J_0(\boldsymbol{\theta})$, which can be addressed by means of a stochastic approximation (SA) algorithm[44], involving the following steps:

1. Initialize $\boldsymbol{\theta}$
2. Obtain an estimated gradient $\widehat{\nabla J_0}(\boldsymbol{\theta})$ of J_0 at $\boldsymbol{\theta}$.
3. Determine a step size β
4. Set $\boldsymbol{\theta} = \boldsymbol{\theta} - \beta \widehat{\nabla J_0}(\boldsymbol{\theta})$

The SA algorithm iterates over steps 2-4 until a stopping criterion is met. In our implementation, we used a central difference (Kiefer-Wolfowitz) estimator[44] of the gradient, $\widehat{\nabla J_0}(\boldsymbol{\theta})$. The performance samples were generated from limited previous experience using a sample augmentation strategy similar to previous works[5].

We still need to determine a model for the estimator ξ_θ. For this, it is useful to characterize the structure of the critical value $\xi(\gamma_n)$. From (9), it is clear that, if $J_n(a_j) > E[J_n(\hat{a}_j) | \hat{a}_j \neq a_j^*, \mathbf{o}_n]$, then $\gamma_n > 1$. Given that $Z_n > 0$[43], this implies that $\xi(\gamma_n) < 0$. This condition is intuitive: if moving to a non-optimal $\hat{a}_j$ implies a smaller cost-to-go than remaining at a_j, then the stopping condition must be fulfilled, i.e. $Z_n > \xi(\gamma_n)$ must hold. For this to be true, no matter what is the (positive) value of Z_n, the

Algorithm 1 Local Exploration with Optimal Stopping

Input: $\mathcal{A}$, a_{j_0}, N, ξ_θ, *SelectAction*
$n = 0$, $\mathbf{o}_n = \emptyset$, $Z_n = -\infty$
for $k = 0, 1, 2, \ldots$ **do**
  $t = 1$, continue = true,
  $\mathcal{A}_{j_k} \equiv$ exploration region with a_{j_k} as central action
  while continue **do**
    $n = n + 1$
    obtain action $a \leftarrow$*SelectAction*$(\mathcal{A}_{j_k}, \mathbf{o}_{n-1})$
    obtain observation $F_n(a)$
    if $\mathbf{o}_{n-1}$ contains N observations **then**
      remove oldest $(a', F_n(a'))$ from $\mathbf{o}_{n-1}$
      update $n_{a'}$ and $\hat{\mu}_{a'}$
    end if
    $\mathbf{o}_n = \{\mathbf{o}_{n-1}, (a, F_n(a))\}$
    update n_a and $\hat{\mu}_a$
    $\hat{a}_j = \max_{a \in \mathcal{A}_j} \hat{\mu}_a$
    if $n_a > 0$ for all $a \in \mathcal{A}_{j_k}$ **then**
      update Z_n using (11) and (12)
    end if
    if $(Z_n > \xi_\theta(\mathbf{o}_n))$ or $(t = N)$ **then**
      continue = false
      $a_{j_{k+1}} = \hat{a}_j$
    else
      $t = t + 1$
    end if
  end while
end for

boundary function $\xi(\gamma_n)$ must be negative. By reasoning similarly for the case $J_n(a_j) < E[J_n(\hat{a}_j)|\hat{a}_j \neq a_j^*, \mathbf{o}_n]$ we obtain a basic structural result for the rejection boundary:

$$\begin{aligned} &\xi(\gamma_n) < 0, \text{ if } E[J_n(\hat{a}_j)|\hat{a}_j \neq a_j^*, \mathbf{o}_n] - J_n(a_j) < 0 \\ &\xi(\gamma_n) > 0, \text{ if } E[J_n(\hat{a}_j)|\hat{a}_j \neq a_j^*, \mathbf{o}_n] - J_n(a_j) > 0 \end{aligned} \tag{14}$$

From the above condition it can be checked that, at the state associated to the highest cost, the critical value $\xi(\gamma_n)$ is negative.

In consequence, it is convenient to select a model for ξ_θ satisfying the above properties, e.g. being monotonically increasing with the estimated performance, $\hat{\mu}_{\hat{a}_j}$, starting in negative values at smaller values, and becoming positive at larger ones. The policy resulting from this structure follows a reasonable cautionary principle: the higher the attained performance, the greater the estimation accuracy required to make a transition.

We have selected a model for ξ_θ requiring a relatively small dimension for $\boldsymbol{\theta}$. Since the attainable performance is the main decision criteria, the input of the model is the estimated performance of the best local action $\hat{\mu}_{\hat{a}_j}$, which is obtained from the observation vector $\mathbf{o}_n$. The output is obtained by means of a weighted sigmoid function, as follows

$$\xi_\theta(\mathbf{o}_n) = \xi_\theta(\hat{\mu}_{\hat{a}_j}) = \theta_0 + \frac{\theta_1}{1 + \exp(\theta_3 + \theta_4 \hat{\mu}_{\hat{a}_j})}. \tag{15}$$

The initialization of the parameter vector $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)$ should comply the desired structural features.

5 | ALTERNATIVES: SW-UCB-U AND SW-RL

This section presents two alternative approaches that will be used later in the benchmark evaluations of Section 7. One is an existing bandit algorithm for unimodal, time-changing responses, and the other is a reinforcement learning algorithm. Both of them incorporate three elements previously introduced in LEOS: 1) a sliding window for storing past observations; 2) upper bound estimates on the expected action performances; and 3) the concept of exploration regions in order to efficiently exploit the underlying structure of the rewards.

5.1 | SW-UCB-U

The proposal in[42] and[45] for MAB problems in time-chaning environments uses the following principles: At each time-slot, the algorithm identifies the action with the best empirical performance in $\mathbf{o}_n$ (the *leader*). The *neighborhood* of the leader is the set containing the leader and actions surrounding the leader. This set is conceptually equivalent to the *exploration region* in our approach. For each action in the neighborhood, an index is computed. Two index options are considered in[42], the Kullback-Leibler index and the UCB index. The first option is limited to systems where rewards distributions are within one-parameter exponential families, and is computational more demanding because computing each index requires solving an optimization problem. Therefore, we consider the second option, referred to as SW-UCB-U, which allows us to compare all the algorithms under equal conditions. The action selected at each n is determined as follows:

- Select the leader l if $\frac{n_l-1}{m}$ is an integer, where n_l is the number of times that the leader is present in $\mathbf{o}_n$.
- Otherwise, select the action with the largest index within the leader's neighborhood.

5.2 | SW-RL

As a second benchmark, we consider a conventional model-free RL approach, i.e. one that does not consider the structural aspects exploited by LEOS. At each time-slot n, the RL agent must decide which action to select from the current exploration region, or if a transition should be made to an adjacent region with the best estimated action. The RL algorithm must therefore solve problem (4) without the restriction of using UCB for action selection.

The state space generated by all possible $\mathbf{o}_n$ vectors has $2N$ dimensions, N of which have continuous domains. Tabular RL approaches are therefore unfeasible for this problem. The control space, in contrast, is relatively small, comprising only $m+1$ actions: the m actions in current exploration region $\mathcal{A}_n$ and the transition decision. The control is denoted by u, and is $u=a$ if action $a \in \mathcal{A}_n$ is selected, and $u=0$ if a transition must be made. The control space is denoted by $\mathcal{U}$. Given this setting, policy gradient[46] is a suitable approach. This technique implies the parametrization of the preferences for each state action pair $h(\mathbf{o}_n, u, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a parameter vector. Actions with the highest preferences, are given the highest probabilities of being selected. The softmax distribution is a usual choice for obtaining randomized policies: $P(u|\mathbf{o}_n, \boldsymbol{\theta}) = e^{h(\mathbf{o}_n,u,\boldsymbol{\theta})}(\sum_{u\in\mathcal{U}} e^{h(\mathbf{o}_n,u',\boldsymbol{\theta})})^{-1}$.

The preference functions are obtained from feature vectors $x(\mathbf{o}_n, u)$ by $h(\mathbf{o}_n, u, \boldsymbol{\theta}) = \boldsymbol{\theta}^T x(\mathbf{o}_n, u)$. In our setting the feature vectors $x(\mathbf{o}_n, u)$ comprise the same information used by LEOS and SW-UCB-U: $\hat{\mu}_a$ and $\hat{\psi}_a$ for the actions in $\mathcal{A}_n$. Therefore, for $u=a \in \mathcal{A}_n$ we have $x(\mathbf{o}_n, a) = (\mathbf{1}_{\{n_a=0\}}, \hat{\mu}_a, \hat{\psi}_a, 0)^T$ where $\mathbf{1}_{\{n_a=0\}}$ is an indicator function that equals 1 if a has not been selected during the last N time-slots, and 0 otherwise. For $u=0$, the feature vector is $x(\mathbf{o}_n, u)^T = (0,0,0,1)$. To obtain an effective policy, the parameter vector $\boldsymbol{\theta}$ determining this policy must be learned offline (as in LEOS) by a policy gradient algorithm. For our numerical experiments we have used the REINFORCE algorithm[46].

6 | PERFORMANCE ANALYSIS

This section presents performance bounds for LEOS in terms of regret and expected convergence time for a given system response $\mathcal{F}_n$. The analysis relies on a Markov model of the algorithm comprising two worst-case assumptions, the first one regarding the location of a^*, and the second one related to the transition probabilities of the Markov chain. We show that, in expectation, LEOS approaches a^* at least as fast as the worst-case model. Let us start by defining the Markov model without these worst-case assumption.

6.1 | Markov Model for LEOS

Let Y_k be the index of the exploration region visited by LEOS at the k-th exploration stage. Considering $Y_k = j$, LEOS makes a transition to the exploration region $\mathcal{A}_{j'}$ (i.e. $Y_{k+1} = j'$) if the empirical best action is $a_{j'}$ and any of the following disjoint events take place: either $t < N$ and $Z_n > \xi_\theta(\mathbf{o}_n)$, or $t = N$. These events are determined by the observation vector $\mathbf{o}_n$, which is a random variable that depends only on the current exploration region, $\mathcal{A}_{Y_k}$. This property allows us to model the random sequence $\{Y_k\} = Y_0, Y_1, \ldots$ as a Discrete Time Markov Chain (DTMC) with state space $\mathcal{S}$, and the transition probabilities $p_{j,j'} = P(Y_{k+1} = j' | Y_k = j)$, for j and j' in $\mathcal{S}$.

In the one-dimensional case, each exploration region $\mathcal{A}_j$ can have, at most, two neighboring regions: $\mathcal{A}_{j-1}$, if $j > 1$, and $\mathcal{A}_{j+1}$, if $j < S$. Therefore transitions in $\{Y_k\}$ can be made only to neighboring states or leave the state unchanged, i.e. $\{Y_k\}$ is a birth-death process[47]. The steady state probabilities of $\{Y_k\}$ are defined as $\bar{\pi}_j = P(Y_k = j)$ for $j \in \mathcal{S}$.

The duration, in time-slots, of an exploration stage is a random variable in $[m, N]$, whose distribution depends on the current exploration region $\mathcal{A}_j$ and the OS policy. Let $E[t_j]$ denote the expected amount of time-slots spent by LEOS in an exploration stage at $\mathcal{A}_j$ (i.e., when $Y_k = j$). The average duration of an exploration stage is given by $E[t] = \sum_{j \in \mathcal{S}} \bar{\pi}_i E[t_j]$.

6.2 | Worst Case Model

Let us first define the *worst case setting* as an action arrangement in which there is one action $a_0 \in \mathcal{A}$ whose distance to a^* equals the diameter of $\mathcal{G}$. In the one dimensional case, this is equivalent to assuming $a^* = \gamma_1$ (or $a^* = \phi_1$). By unimodality, the configuration values (actions) $\gamma_1, \gamma_2, \ldots, \gamma_M$ are in strictly decreasing performance order, i.e. $\mu_1 > \mu_2 > \ldots > \mu_M$. The exploration areas are numbered in the same order, thus $\mathcal{A}_1$ denotes the area containing a^*, and $\mathcal{A}_S$ is the area with the worst performing actions. From now on, $\{Y_k\}$ will represent the sequence of visited areas in a worst case setting for the one-dimensional case. The extension to the two-dimensional case is straightforward and omitted for sake of notation clarity.

The *worst case model*, comprises the following features: 1) the actions are arranged according to the worst case setting, 2) the estimation error probability is set to an upper bound $p_e \geq e_n$, and 3) estimation errors always imply making a transition to the exploration area associated to the worst performing action. The sequence of areas visited by the worst case model follows a DTMC denoted by $\{X_k\}$, characterized by the following transition probabilities:

$$\begin{aligned} P(X_k = j+1 | X_{k-1} = j) &= p_e \qquad \text{for } 1 \leq j \leq S-1 \\ P(X_k = j-1 | X_{k-1} = j) &= 1 - p_e \quad \text{for } 2 \leq j \leq S \\ P(X_k = S | X_{k-1} = S) &= p_e \\ P(X_k = 1 | X_{k-1} = 1) &= 1 - p_e \end{aligned} \tag{16}$$

It is straightforward to check that the above probabilities are bounds for the transition probabilities of $\{Y_k\}$, i.e., $p_{j,j-1} \geq (1 - p_e)$ for $S \geq j > 1$, $p_{j,j} + p_{j,j+1} \leq p_e$ for $S > j > 1$, $p_{1,1} \geq (1 - p_e)$, and $p_{S,S} \leq p_e$. These inequalities allow us to use $\{X_k\}$ as a worst case model of $\{Y_k\}$.

The steady-state probabilities $\pi_1, \ldots, \pi_S$ of $\{X_k\}$ are the unique solution to the balance equations:

$$\begin{aligned} \pi_1 &= (1 - p_e)\pi_1 + (1 - p_e)\pi_2 \\ \pi_j &= p_e \pi_{j-1} + (1 - p_e)\pi_{j+1}, \text{ for } 2 \leq j \leq S-1 \\ \pi_S &= p_e \pi_{S-1} + p_e \pi_S \end{aligned} \tag{17}$$

and the normalization equation $\sum_{j=1}^{S} \pi_j = 1$. Defining $\alpha = \frac{p_e}{1-p_e}$ the balance equations (17) can be reduced to the relation $\pi_j = \alpha^{j-1}\pi_1$ for $j = 1, \ldots, S$, which in combination with the normalization condition gives

$$\pi_1 = \left(\sum_{j=1}^{S} \alpha^{j-1} \right)^{-1} = \frac{1 - \alpha}{1 - \alpha^S} \tag{18}$$

The following result shows that the above probability provides a lower bound to the long-term fraction of time (exploration stages) that $\{Y_k\}$ spends in region 1. This bound is related to M and m, since these parameters determine S by $S = \left\lfloor \frac{2(M-m+1)}{m-1} \right\rfloor + 1$, which can be easily derived from the definition of exploration regions for the one-dimensional case.

Lemma 1. Let $\overline{\pi}_j$ and π_j, for $j \in \mathcal{S}$ be the steady state probabilities of $\{Y_k\}$ and $\{X_k\}$ respectively. The following inequality holds: $\overline{\pi}_1 \geq \pi_1$.

Proof. See Appendix A.1.

6.3 | Performance Bounds

This subsection presents upper and lower bounds on the main performance metrics of our proposal. For notation clarity let us define $\Delta_a = \mu^* - \mu_a$ as the expected difference between the performances of the best action and action $a \in \mathcal{A}$. Next result characterizes the long-term average regret per sample.

Theorem 1. The average regret per sample of the LEOS algorithm in the worst case setting has the following upper bound

$$\lim_{n\to\infty} \frac{R(n)}{n} \leq (1 - \pi_1 C) \sum_{a\in\mathcal{A}_S} \frac{\Delta_a}{m} + \pi_1 C \sum_{a\in\mathcal{A}_1} \frac{\Delta_a}{m} \tag{19}$$

where $C = \frac{E[t_1]}{E[t]}$. The average regret per sample has the following lower bound

$$\lim_{n\to\infty} \frac{R(n)}{n} \geq \frac{1}{N} \sum_{a\in\mathcal{A}_1} \Delta_a \tag{20}$$

Proof. See Appendix A.2.

It can be checked that the upper bound diminishes for higher values of π_1, attaining its minimum for $\pi_1 = 1$, which corresponds to a perfect error estimation, $p_e = 0$ (in fact, it suffices to have perfect estimation at $\mathcal{A}_1$). Similarly, a larger ratio $\frac{E[t_1]}{E[t]}$ is also desirable, which is consistent with our structural description of the OS policy. Note that, to some extent, the sample budget N influences the identification error, since allowing more observations per exploration stage, results in higher estimation accuracy.

The reason why the lower bound is greater than 0 is because, even when the algorithm reaches the region of the best action, all local actions within this region must be resampled at least once every N time-slots. This feature is necessary for adapting to changes in the system response $\mathcal{F}_n$, allowing the algorithm to detect when the best action is no longer in the current region and should be sought in a different one. Therefore, it is the cost of the minimal level of exploration required to track the best action and operate in non-stationary environments. The numerical results in the Section 7 confirm that, when $\mathcal{F}_n$ does not change with n, $R(n)/n$ approaches 0 for $N \to \infty$, but not when $\mathcal{F}_n$ changes with n.

An important feature of the algorithm is its responsiveness, i.e. its ability to reach the local region containing the best action from any initial location. The following theorem provides an upper bound on the expected time to reach this region.

Theorem 2 (bound on the expected convergence time). Let T denote the (random) amount of samples required by the LE algorithm to reach $\mathcal{A}_1$ for the first time. In the worst case setting, the following inequality holds

$$E[T] \leq \frac{N}{(1-p_e)} \sum_{j=1}^{S-2} \frac{1-\alpha^{j+1}}{1-\alpha}. \tag{21}$$

Proof. See Appendix A.3.

Finding a lower bound for $E[T]$ is straightforward. Assuming $X_0 = S$, it takes at least $(S-1)$ transitions to reach region 1, therefore $E[T] \geq m(S-1)$. Obviously, these bounds indicate that the larger the state space (S), the slower the convergence. Additionally, the upper bound suggests that increasing the sample budget N might also slow down convergence. These claims have been validated by our simulation experiments.

7 | NUMERICAL RESULTS

7.1 | Simulation Methodology

We have obtained a complete system response $\mathcal{F}_n$ for a cluster of LTE eNBs, following the 3GPP guidelines for LTE performance evaluation[39]. The simulated network layout consists of a hexagonal grid of 19 three-sectorial macro eNBs (57 macro cells), and 12 pico cells overlapping each macro cell. The simulations are executed for the inner 21 macro cells, while the remaining 36 outer macro cells emulate the interference of a larger network. The total interference at each terminal is the aggregate of the signals transmitted from all nearby base stations, i.e. the four closest macro eNBs and all the picos within the same macro cell.

Our simulator generates user traffic following the 3GPP FTP traffic model[39]. According to this model, each arrival implies the download of one file of 0.5 Mbytes, at a random location within the macro cell. In particular, the arrival falls within the coverage area of a pico cell (with probability 2/3), or uniformly over the whole macro cell (with probability 1/3). The overall arrival rate for each macro cell is $\lambda = 37.5$ files per second. Each downloaded file results in one user throughput sample.

The sizes of the ABS ratio and CRE bias configuration sets are $|\Gamma| = 7$ and $|\Phi| = 10$ respectively, resulting in $M = 70$ total actions. For each configuration $a \in \mathcal{A}$, the simulator generates 40 independent samples of $F_n(a)$ to obtain estimations of

Network layout	19 macro eNBs, 57 directional macro cells, 500 m Inter-Site Distance, 12 pico eNBs per sector
System Bandwidth	10 MHz
Frame duration	Subframe 1 ms, Protected-subframe pattern 8 ms, Frame 10 ms
Transmit power	Macro cell 46 dBm, pico cell 30 dBm
Antenna Pattern (macro sector)	$A_H(\phi) = -\min\left[12\left(\frac{\phi}{\phi_{3\text{dB}}}\right)^2, A_m\right]$, $\phi_{3\text{dB}} = 70$ degrees $A_m = 25$ dB
Antenna Pattern (pico)	Omnidirectional
Antenna gains	macro: 14 dBi; micro: 5 dBi
Macro to UE path loss	$128.1 + 37.6 \cdot \log_{10}(R[\text{Km}])$ where R is the macro eNB to UE distance
Pico to UE path loss	$149.7 + 36.7 \cdot \log_{10}(R[\text{Km}])$ where R is the pico cell to UE distance
Shadow fading	Lognormal distribution with 10 dB standard deviation
Thermal Noise	-174 dBm/Hz
Scheduling Algorithm	Proportional Fair (PF)
Traffic Model	File Transfer Protocol (FTP)
File size	0.5 Mbytes
λ [UEs/s] (Offered traffic load) [Mbps])	37.5 UEs/s (150 Mbps)
Minimum distances	Macro - pico: 70 m; Macro - UE: 35 m; Pico - pico: 40 m; Pico - UE: 10 m

TABLE 3 Simulation Parameters

the mean $\hat{\mu}_a$ and the variance $\hat{\sigma}_a^2$ of $F_n(a)$. Each sample of $F_n(a)$ is the 5$^{\text{th}}$ percentile of 1000 user throughput samples. In our scenario of 21 macro cells, each one generating an average of 37.5 throughput samples per second, this means that the system obtains one $F_n(a)$ sample every 1.27 seconds. Note that, in a real deployment, this sampling rate could be made smaller and synchronous by sampling at the frame level, instead of the application level. Note also the small signalling overhead introduced. Assuming that a throughput sample is encoded in 16 bits, gathering the throughput data in the central controller would only require an additional bandwidth of $37.5 \times 21 \times 16 = 12.6$ Kbit/s.

Following the procedure in[5] we generate new $F_n(a)$ samples at each a by generating Gaussian random variables characterized by $\hat{\mu}_a$ and $\hat{\sigma}_a^2$. Both LEOS and SW-RL can be trained offline using this sample augmentation method.

The initial configuration at every training episode was the most distant action with respect to a^*. The comparative evaluation of the algorithms was done under non-stationary conditions, which were emulated by slightly changing the system response every p time-slots. We refer to p as the *change period*, and $f = 1/p$ is the *change rate*. Each change implies a random displacement of the best action a^* to one of its adjacent regions, preserving the unimodal structure of the system response.

7.2 | LEOS Configuration

The theoretical results for LEOS anticipated that the configuration of the sample budget N must face a tradeoff between regret and convergence time in stationary conditions. We have assessed this tradeoff numerically, by measuring the regret per sample $R(n)/n$ and the expected convergence time $E[T]$ for different values of N for a given $\mathcal{F}_n$. Each performance measurement has been obtained by averaging the results of 500 simulation runs of 20000 samples each. The results, shown in Figure 5 , confirm that larger values of N provide higher estimation accuracy, yielding smaller values of the long-term regret, but resulting in longer exploration stages and thus slower convergence.

In time changing conditions, minimizing the regret requires a sufficiently accurate estimation, but also adaptability to changes in $\mathcal{F}_n$. If the location of a^* changes, the algorithm should be able to quickly attain this new location, for which it needs a sufficiently small $E[T]$. This suggests that there should be an optimal N value for each change rate f, attaining the optimal balance between estimation accuracy and responsiveness. Figure 6 shows the average regret per sample versus N, for three different change periods, 200, 350 and 500. The best configuration of N in LEOS was $N = p = 1/f$ for all the change rates used in this section.

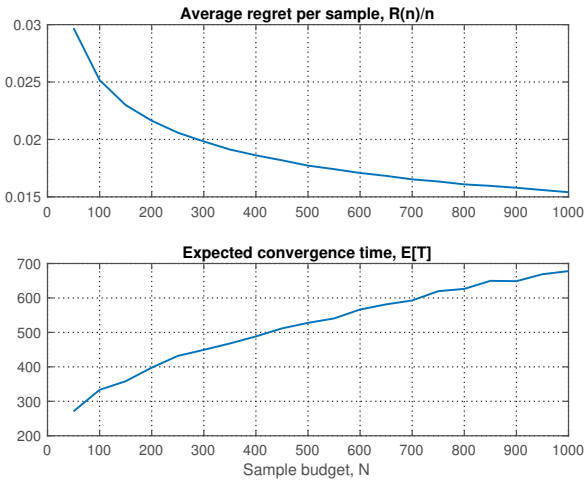

FIGURE 5 Average regret per sample $R(n)/n$ and expected convergence time $E[T]$ (in time-slots) of LEOS in a stationary environment versus the sample budget N.

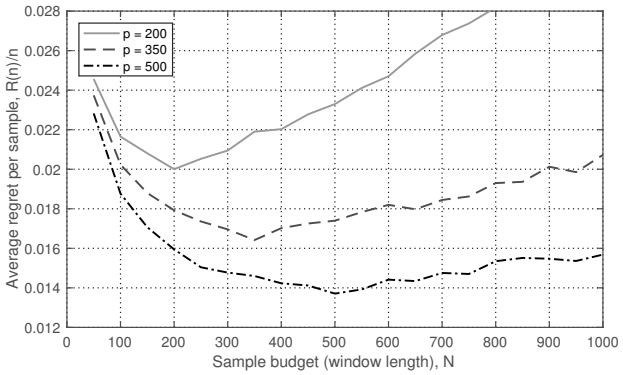

FIGURE 6 Average regret per sample of LEOS versus N under 3 different change periods for a time-varying $\mathcal{F}_n$.

7.3 | Benchmark Evaluation

In this subsection we compare our proposed algorithm with the alternative approaches SW-UCB-U and SW-RL. Figure 7 shows the average regret per sample $R(n)/n$ over time for each algorithm under stationary conditions ($\mathcal{F}_n$ remains unchanged during every simulation run). Each curve was obtained by averaging 100 independent simulation runs. This figure also includes the results of a bandit algorithm (UCB) operating on the whole action space $\mathcal{A}$. Compared to the structure-aware algorithms, the regret of a conventional bandit approach is more than one order of magnitude higher, which highlights the importance of exploiting the underlying structure of the rewards.

Figure 8 shows how $R(n)/n$ evolves over time for change periods $p = 500$ and $p = 200$. Each algorithm is configured at its best performing window size, which is $N = p$ for both LEOS and SW-RL, and $N = 2p$ for SW-UCB-U. We can see that LEOS obtains the smallest $R(n)/n$ both in the early stages and in the long term, and more importantly, we observe that the performance attained by LEOS when $\mathcal{F}_n$ changes over time, is comparatively better than the alternatives. In order to assess how the changing rate influences the performance, we have estimated the average regret at the end of 100 independent runs of 20000 samples each one, for change periods ranging from 100 to 1000 time-steps. Figure 9 summarizes the results obtained, showing that for higher change rates (smaller change periods), the performance gap of LEOS is higher.

The histograms of the selection frequency of each action, provide further insight on the algorithm performance. Figure 10 shows the histograms for a change period of 200 time-slots, with the actions arranged in the x-axes by decreasing performance order. Because of the variability of $\mathcal{F}_n$, the best action is not the one selected most frequently by any algorithm (in contrast to

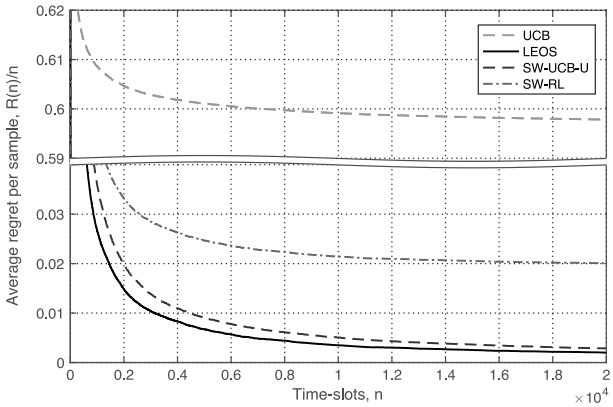

FIGURE 7 Average regret per sample $R(n)/n$ versus n for UCB, LEOS, SW-UCB-U, and SW-RL in a stationary environment.

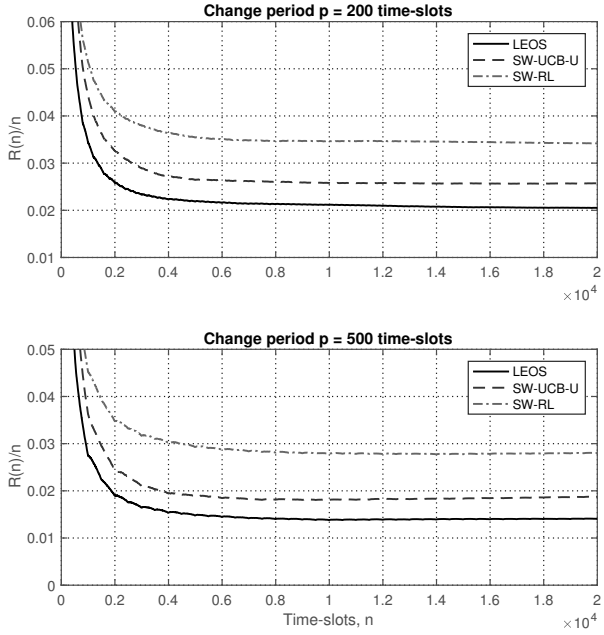

FIGURE 8 Average regret per sample $R(n)/n$ versus n for LEOS, SW-UCB-U, and SW-RL for change rates $1/200$ and $1/500$.

the behavior under stationary conditions). The sampling frequency in LEOS is more concentrated in lower numbered actions compared to the other algorithms. The superior tracking capabilities of LEOS are related to the fact that, in contrast to SW-UCB-U, it leverages past experience when making transition decisions thanks to the offline learning phase. Summarizing, in our experiments under time-varying conditions, LEOS obtained a long-term regret per sample that was, in average, 22% lower than SW-UCB-U, and 48% lower than SW-RL. The performance gap between LEOS and SW-UCB-U is proportional to the change rate of $\mathcal{F}_n$, becoming larger at faster rates.

8 | CONCLUSIONS

This paper presented a novel online reinforcement learning algorithm, LEOS, for finding efficient configurations of the interference coordination parameters in LTE HetNets. The key idea of our proposal is to consider each local exploration stage as an

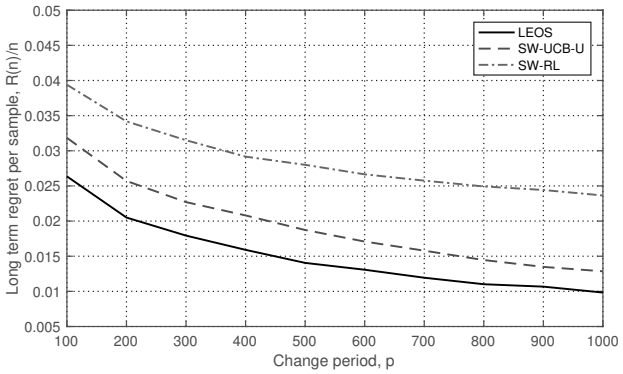

FIGURE 9 Average regret per sample $R(n)/n$ after $n = 20000$ time-steps versus the change period p, for LEOS, SW-UCB-U, and SW-RL.

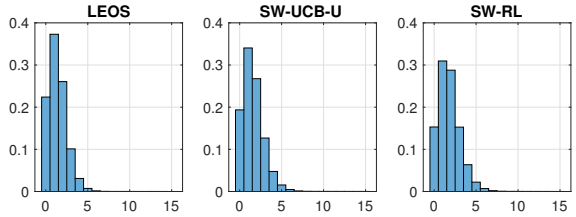

FIGURE 10 Histograms with the estimated action selection frequency for $p = 200$. The actions are ordered by decreasing performance.

optimal stopping (OS) problem aimed at minimizing the long-term regret by making decisions on whether to continue or to end current stage. We show that solving the OS problem is equivalent to adjusting the estimation accuracy of a sequential likelihood ratio test (LRT) aimed at identifying the best local action. Consequently, our solution strategy is based on learning the parameters of an effective decision boundary for the ongoing LRT. We consider two alternatives: a bandit algorithm for non-stationary environments with unimodal structre (SW-UCB-U) and a policy gradient RL algorithm (SW-RL). Our numerical results show that, in a time-varying scenario, LEOS clearly outperforms both SW-UCB-U and SW-RL within a wide range of change rates.

Our proposal is flexible enough to integrate future extensions and enhancements. We would like to discuss two potential improvements. First, to incorporate additional information into the decision process. For example, the agent could observe the user traffic load in the system and map it into a predefined set of load levels. The current load level can be used as a *context*, allowing the learning process to be conducted in a contextual bandit way. In this setting, each context would be associated to its own instance of the LEOS agent (with its own history, local exploration region, etc). Second, to adjust the CRE bias parameter of each small cell by considering the CRE bias provided by LEOS as a reference value to which each small cell would add its own offset. Each station would learn which offset is better for each CRE reference value by solving a contextual bandit problem. Note that the non-stationarity introduced by the distributed learning of the offsets would not be a problem for the centralized learning process, since LEOS is designed specifically for operating under time varying conditions.

8.1 | Acknowledgements

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

References

1. Acharya J, Gao L, Gaur S. *Heterogeneous Networks in LTE-advanced.* John Wiley & Sons . 2014.

2. Tech. Spec. 36.300 v10.5.0 . Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description (Release 10). tech. rep., 3rd Generation Partnership Project (3GPP); 2011.

3. Liu L, Zhou Y, Vasilakos AV, Tian L, Shi J. Time-domain ICIC and optimized designs for 5G and beyond: a survey. *Science China Information Sciences* 2019; 62(2): 21302.

4. Soret B, De Domenico A, Bazzi S, Mahmood NH, Pedersen KI. Interference coordination for 5G new radio. *IEEE Wireless Communications* 2017; 25(3): 131–137.

5. Ayala-Romero JA, Alcaraz JJ, Vales-Alonso J. Data-driven configuration of interference coordination parameters in HetNets. *IEEE Transactions on Vehicular Technology* 2018.

6. Deb S, Monogioudis P, Miernik J, Seymour JP. Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets. *IEEE/ACM Transactions on Networking* 2014; 22(1): 137–150.

7. Liu A, Lau VK, Ruan L, Chen J, Xiao D. Hierarchical radio resource optimization for heterogeneous networks with enhanced inter-cell interference coordination (eICIC). *IEEE Transactions on Signal Processing* 2014; 62(7): 1684–1693.

8. Bin Sediq A, Schoenen R, Yanikomeroglu H, Senarath G. Optimized Distributed Inter-cell Interference Coordination (ICIC) Scheme using Projected Subgradient and Network Flow Optimization. *IEEE Transactions on Communications* 2015; 63(1): 107–124.

9. Vasudevan S, Pupala RN, Sivanesan K. Dynamic eICIC-A proactive strategy for improving spectral efficiencies of heterogeneous LTE cellular networks by leveraging user mobility and traffic dynamics. *IEEE Transactions on Wireless Communications* 2013; 12(10): 4956–4969.

10. Ali MS, Coucheney P, Coupechoux M. Load Balancing in Heterogeneous Networks Based on Distributed Learning in Near-Potential Games. *IEEE Transactions on Wireless Communications* 2016; 15(7): 5046–5059.

11. 3GPP R1-100142 . System performance of heterogeneous networks with range expansion. tech. rep., 3rd Generation Partnership Project (3GPP); 2010.

12. Lattimore T, Szepesvári C. Bandit algorithms. *preprint* 2018: 28.

13. Ayala-Romero JA, Alcaraz JJ, Vales-Alonso J, Egea-Lopez E. Online Optimization of Interference Coordination Parameters in Small Cell Networks. *IEEE Transactions on Wireless Communications* 2017; 16(4): 6635–6647.

14. Al-Rawi M. A dynamic approach for cell range expansion in interference coordinated LTE-advanced heterogeneous networks. In: IEEE. ; 2012: 533–537.

15. Mishra S, Sengupta A, Murthy CSR. Enhancing the performance of HetNets via linear regression estimation of Range Expansion Bias. In: IEEE. ; 2013: 1–6.

16. Simsek M, Bennis M, Güvenç I. Learning based frequency-and time-domain inter-cell interference coordination in HetNets. *IEEE Transactions on Vehicular Technology* 2015; 64(10): 4589–4602.

17. Daeinabi A, Sandrasegaran K. A fuzzy Q-learning approach for enhanced intercell interference coordination in LTE-Advanced heterogeneous networks. In: IEEE. ; 2014: 139–144.

18. Soret B, Pedersen K, others . Centralized and Distributed Solutions for Fast Muting Adaptation in LTE-Advanced HetNets. *IEEE Transactions on Vehicular Technology* 2015; 64(1): 147–158.

19. Al-Rawi M, Huschke J, Sedra M. Dynamic protected-subframe density configuration in LTE heterogeneous networks. In: Munich (Germany). ; July 2012: 1–6.

20. Wang YC, Huang CC. Efficient management of interference and power by jointly configuring ABS and DRX in LTE-A HetNets. *Computer Networks* 2019; 150: 15–27.

21. Wang YC, Huang BJ. Efficient Coordination of Almost Blank Subframes with Coupling Macro-cells in Heterogeneous Networks. *International Journal of Communication Systems* 2019.

22. Pedersen K, Soret B, Barcos S, Pocovi G, Wang H. Dynamic Enhanced Inter-Cell Interference Coordination for Realistic Networks. *IEEE Transactions on Vehicular Technology* 2016; 65(7): 5551–5562.

23. Iacoboaiea OC, Sayrac B, Jemaa SB, Bianchi P. SON Coordination in Heterogeneous Networks: A Reinforcement Learning Framework. *IEEE Transactions on Wireless Communications* 2016; 15(9): 5835–5847.

24. Zheng K, Yang Z, Zhang K, Chatzimisios P, Yang K, Xiang W. Big data-driven optimization for mobile networks toward 5G. *IEEE Network* 2016; 30(1): 44–51.

25. He Y, Yu FR, Zhao N, Yin H, Yao H, Qiu RC. Big data analytics in mobile cellular networks. *IEEE Access* 2016; 4: 1985–1996.

26. Bi S, Zhang R, Ding Z, Cui S. Wireless communications in the era of big data. *IEEE Communications Magazine* 2015; 53(10): 190–199.

27. Mudassir A, Akhtar S, Kamel H, Javed A. Intelligent spectral efficiency and energy efficiency enhancement in LTE-Advanced heterogeneous networks. *Transactions on Emerging Telecommunications Technologies* 2018; 29(10): e3431.

28. Lynch D, Fenton M, Fagan D, Kucera S, Claussen H, O'Neill M. Automated Self-Optimization in Heterogeneous Wireless Communications Networks. *IEEE/ACM Transactions on Networking* 2019; 27(1): 419–432.

29. Jin W, Huilin J, Zhiwen P, Nan L, Xiaohu Y, Tianle D. Joint user association and ABS proportion optimization for load balancing in HetNet. In: Nanjing (China). ; October 2015: 1–6.

30. Sung DH, Baras JS. Utility-based almost blank subframe optimization in heterogeneous cellular networks. In: IEEE. ; 2014: 3622–3627.

31. Trabelsi N, Roullet L, Feki A. A Generic Framework for Dynamic eICIC Optimization in LTE Heterogeneous Networks. In: IEEE. ; 2014: 1–6.

32. Li J, Wang X, Li Z, Wang H, Li L. Energy Efficiency Optimization Based on eICIC for Wireless Heterogeneous Networks. *IEEE Internet of Things Journal* 2019.

33. Hajijamali Arani A, Omidi MJ, Mehbodniya A, Adachi F. A distributed learning–based user association for heterogeneous networks. *Transactions on Emerging Telecommunications Technologies* 2017; 28(11): e3192.

34. Li L, Zhou Z, Sun S, Wei M. Distributed optimization of enhanced intercell interference coordination and resource allocation in heterogeneous networks. *International Journal of Communication Systems* 2019; 32(6): e3915.

35. Cierny M, Wang H, Wichman R, Ding Z, Wijting C. On number of almost blank subframes in heterogeneous cellular networks. *IEEE Transactions on Wireless Communications* 2013; 12(10): 5061–5073.

36. Wang Y, Ji H, Zhang H. Spectrum-efficiency enhancement in small cell networks with biasing cell association and eICIC: An analytical framework. *International Journal of Communication Systems* 2016; 29(2): 362–377.

37. Kim R, Kim Y, Yu NY, Kim SJ, Lim H. Online Learning-Based Downlink Transmission Coordination in Ultra-Dense Millimeter Wave Heterogeneous Networks. *IEEE Transactions on Wireless Communications* 2019; 18(4): 2200–2214.

38. Soret B, Pedersen KI. Macro transmission power reduction for hetnet co-channel deployments. In: Anaheim (California, USA). ; December 2012: 4126–4130.

39. 3GPP TR 36.814 . Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA Physical Layer Aspects. tech. rep., 3rd Generation Partnership Project (3GPP); 2010.

40. Bubeck S, Cesa-Bianchi N, others . Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 2012; 5(1).

41. Garivier A, Moulines E. On Upper-Confidence Bound Policies for Switching Bandit Problems. In: Springer. ; 2011: 174–188.

42. Combes R, Proutiere A. Unimodal Bandits: Regret Lower Bounds and Optimal Algorithms. In: ; 2014: 521–529.

43. Garivier A, Kaufmann E. Optimal best arm identification with fixed confidence. In: ; 2016: 998–1027.

44. Rubinstein RY, Kroese DP. *Simulation and the Monte Carlo method*. 10. John Wiley & Sons . 2016.

45. Combes R, Proutiere A. Dynamic Rate and Channel Selection in Cognitive Radio Systems. *IEEE Journal on Selected Areas in Communications* 2015; 33(5): 910–921.

46. Sutton RS, Barto AG. *Reinforcement learning: An introduction*. MIT press . 2018.

47. Bertsekas D, Tsitsiklis J. *Introduction to Probability*. Athena Scientific, Belmont, MA . 2002.

48. Ross SM. *Stochastic Processes. Second Edition*. Wiley, New York . 1996.

How to cite this article: J.J. Alcaraz, J.A. Ayala-Romero, J. Vales-Alonso, and F. Losilla-Lopez (2020), Online Reinforcement Learning for Adaptive Interference Coordination, *Trans. Emerging Tel. Tech.*, *To Be Added*.

APPENDIX

A PROOFS

A.1 Proof of Lemma 1

Because $\{Y_k\}$ is a birth death process, the *local balance* equations hold ($\overline{\pi}_j p_{j,j+1} = \overline{\pi}_{j+1} p_{j+1,j}$ for $j = 1, \dots, S$, which leads to $\overline{\pi}_j = \overline{\alpha}_j \overline{\pi}_1$, for $j = 2, \dots, S$, where $\overline{\alpha}_j = \prod_{j'=2}^{j} \frac{p_{j'-1,j'}}{p_{j',j'-1}}$. Similarly, for $\{X_k\}$ we have $\pi_j = \alpha^{j-1}\pi_1$, where $\alpha = \frac{p_e}{(1-p_e)}$. Using the normalization condition we have

$$\begin{aligned} \overline{\pi}_1 &= \left(1 + \sum_{j=2}^{S} \overline{\alpha}_j\right)^{-1} \\ \pi_1 &= \left(1 + \sum_{j=2}^{S} \alpha^{j-1}\right)^{-1} \end{aligned} \tag{A1}$$

Because the transition probabilities of $\{X_k\}$ are bounds to those of $\{X_k\}$, we have that $\prod_{j'=2}^{j} p_{j'-1,j'} \leq p_e^{j-1}$, and $\prod_{j'=2}^{j} p_{j',j'-1} \geq (1-p_e)^{j-1}$ and then

$$\overline{\alpha}_j = \prod_{j'=2}^{j} \frac{p_{j'-1,j'}}{p_{j',j'-1}} \leq \frac{p_e^{j-1}}{(1-p_e)^{j-1}} = \alpha^{j-1} \tag{A2}$$

for $j = 2, \dots, S$. Therefore $\sum_{j=2}^{S} \overline{\alpha}_j \leq \sum_{j=2}^{S} \alpha^{j-1}$, which applied in (A1) yields $\overline{\pi}_1 \geq \pi_1$. □

A.2 Proof of Theorem 1

The sequence of instantaneous regret values r_n, for $n = 1, 2, \dots$ can be divided into consecutive groups corresponding to the observations taken at each exploration stage. The counting process associated to the exploration stages $k = 0, 1, 2, \dots$ can be seen as a renewal process in which the time between renewals is a random variable determined by the OS policy, and $E[t]$ is the expected time (in time-slots) between renewals. Similarly let $E[R]$ denote the expected accumulated regret during an exploration stage, and let $E[R_j] = E[R|Y_n = j]$ denote the expected accumulated regret during an exploration stage at $\mathcal{A}_j$. Given these elements, we have a renewal reward process in which the expected reward received upon each renewal is $E[R]$. By Theorem 3.6.1. of[48] we have that

$$\lim_{n\to\infty} \frac{R(n)}{n} = \frac{E[R]}{E[t]} \tag{A3}$$

Let us start with the upper bound. Because $\{Y_k\}$ is a Markov process, the sequence of accumulated rewards per stage is Markov reward process, in which the expected reward associated to each state j is $E[R_j]$. Therefore, we can express $E[R]$ as $E[R] = \sum_{j\in S} \overline{\pi}_j E[R_j]$ where $\overline{\pi}_j$ is the steady state probability of $\{Y_k\}$. Because LEOS uses a regret minimization algorithm (UCB),

$E[R_j]$ is upper bounded by the accumulated regret of an algorithm selecting actions at random, with uniform probability: $E[R_j] \leq E[t_j] \sum_{a\in\mathcal{A}_j} \frac{\Delta_a}{m}$, therefore we have

$$\begin{aligned} \frac{E[R]}{E[t]} &\leq \sum_{j\in S} \sum_{a\in\mathcal{A}_j} \overline{\pi}_j \frac{E[t_j]}{E[t]} \frac{\Delta_a}{m} \\ &= \sum_{j\neq 1} \sum_{a\in\mathcal{A}_j} \overline{\pi}_j \frac{E[t_j]}{E[t]} \frac{\Delta_a}{m} + \sum_{a\in\mathcal{A}_1} \overline{\pi}_1 \frac{E[t_1]}{E[t]} \frac{\Delta_a}{m} \end{aligned} \tag{A4}$$

Note that $\overline{\pi}_j \frac{E[t_j]}{E[t]}$ is the probability that the LE algorithm is exploring region j at a given time-slot n, $P[_{a\in\mathcal{A}_j}]$. To show this, we can apply the renewal reward theorem considering renewal periods with average length $E[t]$ as in (A3), and the reward given by an indicator function $\mathbf{1}_{\{a\in\mathcal{A}_j\}}$, which equals 1 when the selected action belongs to $\mathcal{A}_j$, and 0 otherwise: $P[_{a\in\mathcal{A}_j}] = \lim_{n\to\infty} \frac{E[\mathbf{1}_{\{a\in\mathcal{A}_j\}}]}{n}$ $= \frac{\overline{\pi}_j E[t_j]}{E[t]}$. Using this definition in (A4) we have

$$\begin{aligned} \frac{E[R]}{E[t]} &\leq \sum_{j\neq 1} \sum_{a\in\mathcal{A}_j} P[_{a\in\mathcal{A}_j}] \frac{\Delta_a}{m} + \sum_{a\in\mathcal{A}_1} P[_{a\in\mathcal{A}_1}] \frac{\Delta_a}{m} \\ &\leq \sum_{a\in\mathcal{A}_S} \frac{\Delta_a}{m} \sum_{j\neq 1} P[_{a\in\mathcal{A}_j}] + \sum_{a\in\mathcal{A}_1} P[_{a\in\mathcal{A}_1}] \frac{\Delta_a}{m} \\ &= \sum_{a\in\mathcal{A}_S} \frac{\Delta_a}{m} (1 - P[_{a\in\mathcal{A}_1}]) + \sum_{a\in\mathcal{A}_1} P[_{a\in\mathcal{A}_1}] \frac{\Delta_a}{m} \\ &= (1 - \overline{\pi}_1 \frac{E[t_1]}{E[t]}) \sum_{a\in\mathcal{A}_S} \frac{\Delta_a}{m} + \overline{\pi}_1 \frac{E[t_1]}{E[t]} \sum_{a\in\mathcal{A}_1} \frac{\Delta_a}{m} \end{aligned} \tag{A5}$$

Applying Lemma 1, and defining $C = \frac{E[t_1]}{E[t]}$ we get $\frac{E[R]}{E[t]} \leq (1 - \pi_1 C) \sum_{a\in\mathcal{A}_S} \frac{\Delta_a}{m} + \pi_1 C \sum_{a\in\mathcal{A}_1} \frac{\Delta_a}{m}$.

The lower bound to (A3) corresponds to LEOS exploring at $\mathcal{A}_1$ with perfect estimation, and perfect identification of the best action after a single observation from each action. Because the estimations are reset at most after N time-slots, at least one sample must be taken from each action $a \in \mathcal{A}_1$ every N time-slots, with an expected regret Δ_a, resulting in the lower bound $\frac{1}{N} \sum_{a\in\mathcal{A}_1} \Delta_a$, which concludes the proof. □

A.3 Proof of Theorem 2

The proof of Theorem 2 makes use of the following result.

Lemma 2. Consider the LE algorithm operating in the one-dimensional case, over a set of unimodal actions comprising S exploration regions. Assume that the optimal action is located in region 1. Let $X_0, X_1, \ldots$ be the sequence of exploration regions visited by LE at each exploration period $k = 0, 1, \ldots$. Let t_j denote the mean first passage time to region 1 starting in region j ($t_j = E\left[\min\left\{k \geq 0 | X_k = 1\right\} | X_0 = j\right]$). The inequality $t_j > t_{j-1}$ holds for $j = S, \ldots, 2$.

Proof. This Lemma is proven by induction. The mean first passage times are given by the following set of equations:

$$\begin{aligned} t_j &= 1 + \sum_{j'=1}^{S} p_{j,j'} t_{j'}, \text{ for } j = 2, \ldots, S \\ t_1 &= 0 \end{aligned} \tag{A6}$$

Because the LE algorithm can only make transitions among adjacent regions, we have

$$t_S = 1 + p_{S,S-1} t_{S-1} + p_{S,S} t_S \tag{A7}$$

solving for t_S we obtain

$$\begin{aligned} t_S &= \frac{1}{1-p_{S,S}} + \frac{p_{S,S-1}}{1-p_{S,S}} t_{S-1} \\ &= \frac{1}{1-p_{S,S}} + t_{S-1} \\ &> t_{S-1}. \end{aligned} \tag{A8}$$

The relation $t_2 > t_1$ holds trivially, since $t_2 \geq 1 > t_1 = 0$. Now, we assume that $t_{j+1} > t_j$ (for $S > j > 2$), and then show that $t_j > t_{j-1}$. From (A6) we have

$$t_j = 1 + p_{j,j-1} t_{j-1} + p_{j,j} t_j + p_{j,j+1} t_{j+1}. \tag{A9}$$

Using $t_{j+1} > t_j$ we obtain the following inequality

$$\begin{aligned} t_j &> 1 + p_{j,j-1} t_{j-1} + p_{j,j} t_j + p_{j,j+1} t_j \\ &= 1 + p_{j,j-1} t_{j-1} + (1 - p_{j,j-1}) t_j \end{aligned} \tag{A10}$$

Solving for t_j we obtain

$$t_j > \frac{1}{p_{j,j+1}} + t_{j-1} > t_{j-1} \tag{A11}$$

which concludes the proof of Lemma 2. □

Now we can proof Theorem 3. Because LE takes at most N samples from each exploration region during each exploration stage, we have that $E[T|X_0 = j] \leq N t_j$, therefore using Lemma 2 we have that

$$E[T] \leq \max_{j \in S} E[T|X_0 = j] \leq N t_S \tag{A12}$$

Let us consider the system of equations (A6) providing the first passage times. For t_S, we have

$$\begin{aligned} t_S &= 1 + p_{S,S-1} t_{S-1} + p_{S,S} t_S \\ &\leq 1 + (1 - p_e) t_{S-1} + p_e t_S \end{aligned} \tag{A13}$$

where the inequality comes from the fact that $p_e \geq p_{S,S}$ and $t_S > t_{S-1}$ (Lemma 2). For $S > i > 2$, the following inequalities hold

$$\begin{aligned} t_j &= 1 + p_{j,j-1} t_{j-1} + p_{j,j} t_j + p_{j,j+1} t_{j+1} \\ &\leq 1 + p_{j,j-1} t_{j-1} + (p_{j,j} + p_{j,j+1}) t_{j+1} \\ &\leq 1 + (1 - p_e) t_{j-1} + p_e t_{j+1} \end{aligned} \tag{A14}$$

where the first inequality results from Lemma 2. The resulting system of inequalities can be simplified by substitution. Equation (A13) can be expressed as $t_S \leq \frac{1}{1-p_e} + t_{S-1}$, which applied into (A14) for $j = S - 1$ results in $t_{S-1} \leq \frac{1}{1-p_e} + \frac{p_e}{(1-p_e)^2} + t_{S-2}$. Proceeding backwards, we obtain the following set of recursive equations

$$t_{S-j} \leq \frac{1}{1-p} \sum_{k=0}^{j} \alpha^k + t_{S-j-1}, \text{ for } j = 0, \ldots, S-2 \tag{A15}$$

where $\alpha = \frac{p_e}{1-p_e}$. Using (A15) to solve for t_S we obtain

$$\begin{aligned} t_S &\leq \tfrac{1}{1-p_e} \left(\textstyle\sum_{k=0}^{0} \alpha^k + \sum_{k=0}^{1} \alpha^k + \ldots + \sum_{k=0}^{S-2} \alpha^k \right) \\ &= \frac{1}{1-p_e} \sum_{j=0}^{S-2} \sum_{k=0}^{j} \alpha^k \\ &= \frac{1}{1-p_e} \sum_{j=0}^{S-2} \frac{1 - \alpha^{j+1}}{1 - \alpha} \end{aligned} \tag{A16}$$

which applied in (A12) concludes the proof. □