Orchestrating Energy-Efficient vRANs: Bayesian Learning and Experimental Results

Jose A. Ayala-Romero, Andres Garcia-Saavedra, Xavier Costa-Perez *Senior Member, IEEE,* and George Iosifidis

Abstract—Virtualized base stations (vBS) can be implemented in diverse commodity platforms and are expected to bring unprecedented operational flexibility and cost efficiency to the next generation of cellular networks. However, their widespread adoption is hampered by their complex configuration options that affect in a non-traditional fashion both their performance and their power consumption. Following an in-depth experimental analysis in a bespoke testbed, we characterize the vBS power consumption profile and reveal previously unknown couplings between their various control knobs. Motivated by these findings, we develop a Bayesian learning framework for the orchestration of vBSs and design two novel algorithms: (i) BP-vRAN, which employs online learning to balance the vBS performance and energy consumption, and (ii) SBP-vRAN, which augments our optimization approach with *safe* controls that maximize performance while respecting hard power constraints. We show that our approaches are *data-efficient*, i.e., converge an order of magnitude faster than state-of-the-art Deep Reinforcement Learning methods, and achieve optimal performance. We demonstrate the efficacy of these solutions in an experimental prototype using real traffic traces.

Index Terms—Bayesian Learning, Gaussian Processes, Online Learning, Radio Access Networks, Energy efficiency, Green networks, Network Virtualization, Wireless Testbeds

1 Introduction

Virtualization is considered one of the key approaches for bringing cellular networks up to speed with the demanding services they aspire to offer to users [1]. The latest frontier in this endeavor is the development of virtualized Radio Access Networks (vRAN) where legacy base stations (BSs) are replaced by softwarized stacks such as those developed by srsRAN [2] and OpenAirInterface (OAI) [3]. These novel BSs are fully-configurable and can be deployed in different platforms ranging from commodity servers and small embedded devices to moving nodes such as drones [4]. This RAN transformation constitutes a paradigm shift for cellular networks and is expected to offer the much-needed performance flexibility, facilitate the necessary network densification, and reduce significantly their capital and operating expenses [5]. Hence, it is not surprising that we see today numerous industry efforts aiming to build such BS software stacks [2], design fully-open RAN architectures [6], and even conduct extensive field trials [7].

1.1 The problem

Nevertheless, the advent of vRANs raises novel technical challenges since the virtualized base stations (vBSs) differ significantly from their hardware-based legacy BSs. On the one hand, Open RAN solutions (led by the O-RAN alliance) enable vBS to change in real-time a variety of different operation parameters, such as transmission power and modulation schemes, in order to adapt to the volatile network conditions and dynamic user needs. On the other hand, though this certainly provides network operators an unprecedented level of flexibility, it comes at the cost of less predictable performance due to the complex couplings between the high-dimensional space of tunable control knobs and the resulting performance, as we reveal in Sec. 3. The latter is crucial for economic reasons, especially in light of the increasing network densification; but also because vBSs are often expected to operate under tight energy budgets [8] – consider for instance vBS that are supported with batteries or Power-over-Ethernet (PoE) lines. Therefore, existing resource control policies run the risk of under-utilizing this new type of BSs, or rendering vRANs economically unsustainable. It becomes, therefore, clear that in order to unleash the full potential of vRANs we need to answer two key questions:

(i) What is the performance and power consumption characteristics of virtualized BSs?

(ii) How can we optimize their operation using an adaptive and platform-oblivious approach?

In this paper we tackle these questions following a detailed experimental and analytical methodology.

1.2 Our solution

We start by studying the vBSs operation using different hosting platforms and usage scenarios in a customized wireless testbed. Our results shed light on the relationship between performance (throughput), power consumption, and vBS controls such as the modulation and coding schemes (MCS) and spectrum allocation. For instance, we find that the baseband unit (BBU) consumes power comparable to wireless transmissions, and we observe the vBS power consumption and effective throughput being

- *J. A. Ayala-Romero is with Trinity College Dublin.*
- *A. Garcia-Saavedra is with NEC Labs Europe.*
- *X. Costa-Perez is with NEC Labs Europe, i2CAT and ICREA.*
- *G. Iosifidis is with Delft University of Technology.*

affected by the configurations in a non-linear and non-monotonic fashion. These results depend heavily on the hosting platform and underline the difficulties in optimizing the vBS operation. Moreover, we observe that the uplink (UL)-related computations of the vBS stack consume more power and are more sensitive to MCS and SNR variations, than the respective downlink (DL) computations; a finding attributed to the heavier UL decoding. Besides, we measure the vBS power consumption for concurrent UL and DL processing and find it significantly smaller than the total consumption of these operations when executed separately (only UL or only DL). These findings are particularly important since uplink transmissions are needed to support the ever-growing user traffic. Our analysis is centered on energy since it is the bottleneck vBS resource that affects both their computations and transmissions, and which, if not properly controlled, will induce prohibitive costs and environmental consequences as cellular networks become even more pervasive [9].

The take-away message from these extensive measurements (presented in Sec. 3) is that, *unlike legacy BSs*, virtualized BSs have a complex, poly-parametric, and platform-dependent performance and power consumption profile; and this renders traditional control policies inefficient for their management. In order to overcome this obstacle, we propose and evaluate a novel machine learning framework that learns on-the-fly the vBS operational profiles and selects their optimal configuration based on the network needs and power availability or constraints. In particular, we formulate two energy-aware vBS control problems and design learning algorithms to solve them in a robust fashion: (*i*) BP-vRAN (Bayesian optimization for Power consumption in vRANs), which finds a tunable trade-off between performance and power consumption; and (*ii*) SBP-vRAN (Safe Bayesian optimization for Power consumption in vRANs), which maximizes the vBS performance subject to *hard* constraints on power consumption. The former allows operators to balance performance and power expenses, while the latter is crucial for vBS running on power-constrained platforms, e.g., Power-over-Ethernet cells.

Our algorithms are founded on Bayesian optimization theory [10] and Gaussian Processes (GPs) [11]. These tools are appropriate for our problems because, as we show in this paper, they are remarkably *data-efficient*, which is an important requirement in our case given the high-dimensional nature of our context-action space. The GPs model the behavior of the vBS in terms of performance and power consumption, using measurements that are collected in runtime. Accordingly, we use a contextual bandit framework to *explore* the space of vBS configurations and *exploit* the best ones for each context. For the latter, we use the average UL/DL traffic load and SNR values, which we measure over certain time windows as these are determined by the pertinent 3GPP O-RAN specification [6]. The outcome is a non-parametric algorithmic framework that makes minimal assumptions about the system, adapts to user needs and network conditions, and *provably* maximizes the throughput of the system. Furthermore, drawing ideas from *safe* Bayesian optimization [12], [13], the SBP-vRAN algorithm ensures the vBS power constraints are not violated during exploration, hence enables the vBS deployment on energy-constrained platforms. By its design, this framework outperforms other approaches requiring knowledge of the vBS functions [14] or offline data to approximate them [15], and adaptive techniques that do not offer performance guarantees or rely on strict system modeling assumptions [16], [17] (see Sec. 2).

Finally, we perform an extensive evaluation in a customized testbed based on srsRAN [2], and using several tools to measure in real time the vBS power consumption. This is an important step in our study as it allows us to assess the practical efficacy of the proposed learning algorithms. Indeed, we verified that both solutions converge to the optimal vBS configuration in a variety of scenarios. To that end, we also proposed and evaluated several practical enhancements that expedite the algorithms' convergence. Using real traffic traces, we show, step-by-step, how our framework explores the configurations, and how it refrains from violating the power constraints when necessary. We also benchmark our solution with a state-of-the-art Reinforcement Learning (RL) solution. Namely, we implement a Deep Deterministic Policy Gradient (DDPG) algorithm using an actor-critic neural network (NN) architecture [18], and adapted to our contextual bandit problem. We find that our framework is more data-efficient than such state-of-the-art RL approaches which require orders of magnitude more measurements (hence, also more time) to train the NNs. We believe such experimental comparisons contribute to the ongoing discussion about which AI/ML techniques can in practice solve resource orchestration problems in cellular networks.

1.3 Contributions and paper organization

Motivated by the increasing importance and fast-paced deployment of virtualized base stations [2], [6], [7], we revisit the problem of energy-aware resource orchestration in cellular networks. Using a hybrid experimental and theoretical approach, we make the following contributions: In summary, the main contributions of this paper are:

- We built a bespoke wireless testbed and performed an exhaustive experimental study of the power consumption and performance of vBSs, using different hosting platforms, configurations and use cases. Our experiments reveal hitherto-unknown features of this new class of base stations that depart significantly from the energy consumption profile of legacy base stations.
- We developed a non-parametric learning framework to optimize the vBS operation in runtime; and we propose two algorithms for tackling two key problems: *(i)* BP-vRAN, which balances performance and costs; and *(ii)* SBP-vRAN, which maximizes performance subject to hard power consumption constraints. Our framework is based on Bayesian learning techniques, which remain relatively unexplored in communication networks (cf. Sec. 2), and which we extend to account for the network context and also amend them with practical rules in order to be suitable for vRANs.
- Finally, we assess the performance of our algorithms using realistic contexts (network loads and channel dynamics), and compare their performance and data

requirements with a state-of-the-art RL solution. The findings verify that they constitute strong candidates as the next-generation zero-touch vBS control solution. The source code of BP-vRAN and SBP-vRAN and the produced *experimental datasets are publicly available*, aspiring to facilitate the evaluation of other AI/ML solutions for vRAN orchestration.

This paper extends our preliminary conference version [19] with the following contributions:

- We design and implement a customized version of a state-of-the-art deep reinforcement learning algorithm (DDPG) as a benchmark solution. We configure it to efficiently solve both of the problems investigated in this paper.
- We expand our evaluation section to thoroughly compare our solutions, BP-vRAN and SBP-vRAN, against the DDPG algorithm. We evaluate the convergence rate for both cases and assess the performance of a sudden change on the power budget for the second one. We discuss the pros and cons of Bayesian against reinforcement learning NN-based solutions.

Paper Organization. Section 2 discusses the related work and positions our contributions accordingly, and Section 3 presents experimental measurements that bring to the fore the vBS control challenges. In Section 4 we introduce the system model and formulate the two optimization problems. Section 5 follows with the Bayesian-based learning algorithms for solving the problems at hand, and Section 6 presents a series of experiments that validate our approach and compare it with deep-learning algorithms. We conclude in Section 7.

2 Related Work

2.1 Network Optimization & Automated Configuration

The works that optimize resource management in softwarized cellular networks can be classified to: *(i)* those requiring models that relate control variables to performance metrics; *(ii)* model-free approaches that rely on offline training data; and *(iii)* online learning techniques. Interesting examples in *(i)* include [20] which performs rate control to maximize throughput subject to computing capacity; [14] that selects also the MCS and airtime; and [21] that additionally adapts to traffic. Nonetheless, such models are in practice platform/context dependent and unknown. On the other hand, model-free approaches employ machine learning, e.g., Neural Networks, to approximate performance functions [22]. Such approaches are used in network slicing [23], throughput forecasting [15], edge computing [24], etc. Their efficacy is remarkable as long as there are *enough and representative training data*. Otherwise, we need to employ online learning that has been recently used, for instance, to configure video analytic systems [25] and minimize the power consumption and interference among BSs [26]. Similarly, online convex optimization is used for cloud and IoT resource orchestration [27], [28], but requires convex functions; a condition not satisfied here. Another approach is reinforcement learning (RL), used in spectrum management [16], network diagnostics [29], interference coordination [30], and SDN control [31], among others. In this line, [32], [33] optimize the energy efficiency of the network as a function of some parameters such as the resource block allocation, the transmission power, or the amount of network offloading. Compared to [32], not only we are considering more configuration parameters, we are also considering more relevant aspects and dimensions of the problem. Specifically, in [32], they rely on a simplified setup comprised of some communicating blocks using GNU radio instead of a full system, and on an over-simplistic power consumption model given by a linear equation where the circuit power is considered constant. In marked contrast, we do not make any modeling assumption. We rely on real measurements from a full-fledged 3GPP-compliant system, which moreover show that the consumed power of our target object (a virtualized BBU) is highly variable, shows non-linear behavior, and depends on many aspects. In [33], the authors address the problem of offloading and autoscaling in mobile edge computing considering renewable energy. However, the radio access network (RAN), which is the focus of our work, and hence their approach cannot be applied to our problem.

Similarly to RL, contextual bandits have been employed to adjust video streaming rates [34]; configure BS parameters (e.g., handover thresholds) [35], [36]; assign CPU time to virtualized BSs [17]; and control mmWave networks [37], [38]. Here, instead, we combine Gaussian Processes [11] and contextual bandit algorithms [39] to build a *data-efficient* Bayesian optimization framework [10] with *convergence guarantees*. Our approach captures the non-trivial multimodal correlations of configurations (revealed by our experiments) through GPs, and use these perpetually-updated functions to sample the decision space. Our work draws from the seminal CGP-UCB algorithm [39] which is extended to include vRAN-specific context, to optimize throughput and power costs, and to satisfy hard power constraints. This is crucial for vBS which cannot exceed *at any time* their power threshold, e.g., when they are powered over Ethernet.

Despite been very successful in many problems, ranging from the design of experiments to automated machine learning [10], Bayesian learning algorithms to date have not been used in communication networks, with very few exceptions such as [40] that explores the optimal server configuration for big data computing. Our approach aspires to fill this gap by studying experimentally their efficacy on the vRAN orchestration problem. To that end, we also compare them with a state-of-the-art Deep RL solution: Deep deterministic policy gradient (DDPG) algorithm adapted to our contextual bandit setting. Such sophisticated neural-network based solutions have only recently been used in wireless networks (e.g. for traffic scheduling) [17], [41], [42], and, to the best of our knowledge have not been compared against Bayesian optimization approaches.

2.2 Experimental Profiling of vBS Computing & Power Consumption

Clearly, it is imperative to explore experimentally the operation of these new BSs. The early work of [43] studied

the cost savings when pooling the processing operations of multiple BSs, and [44] proposed a similar vRAN architecture and measured 30% processing load reduction. Other studies considered the effect of MCS, bandwidth, and SNR on BBU computing load [45], [46]. In [47] an OAI simulator was used to model the processing time for different configurations, and [17] presented measurements with srsLTE for the impact of traffic. Our experimental analysis builds on these important works and further measures the impact of new context parameters and radio schedulers on throughput, the coupling of uplink and downlink operations, and the vBS power consumption in different scenarios.

Existing power consumption studies for legacy BSs focus on the effect of power amplifier, RF output, and baseband processing. The work [48] introduced the *EARTH* model which relates the RF output power with the supplied power; and [49] considered also the effect of bandwidth. The works [50], [51] proposed similar models for macro and micro BSs, and [52] studied how the packet length affects the CPU power consumption. A detailed model accounting for the different BS components is presented in [53], [54].

To illustrate the power behavior of legacy BS, we rely on the seminal model proposed in [48], where the consumed power (P_{in}) is given by

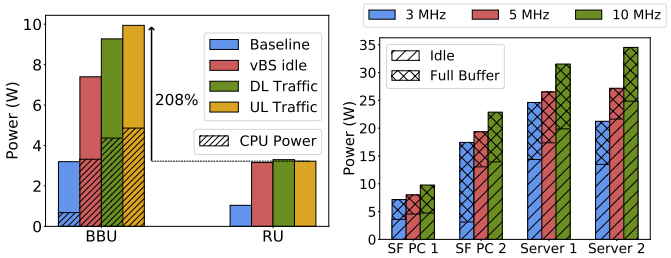
$$P_{\text{in}} = \begin{cases} N_{\text{TRX}} \cdot P_0 + \Delta_{\text{p}} \cdot P_{\text{out}}, & 0 < P_{\text{out}} \leq P_{\text{max}} \\ N_{\text{TRX}} \cdot P_{\text{sleep}}, & P_{\text{out}} = 0 \end{cases} \quad (1)$$

where N_{TRX} is the number of transceivers, P_{out} is the RF output power, P_{max} is the maximum RF output, P_0 represents the power consumption at zero RF output power, P_{sleep} is the power consumption of transceivers components in sleep mode, and Δ_{p} is the slope of the load-dependent power consumption.

Note that the model in eq. (1) is basically focused on the downlink, which is the predominant factor in legacy BSs. Conversely, for the new generation of small form-factor vBSs the uplink and the configuration parameters are equally important[1]. Moreover, although the downlink transmission power and airtime can be captured by P_{out}, other factors such as the MCS and channel quality are not considered in eq. (1) and we have found they are relevant in the consumed power of vBSs. We observe that the model in eq. (1) is linear, which is a good approximation of the measurements in [48]. Its slope, given by Δ_{p}, characterizes the relation between the consumed power and P_{out} the total RF output power radiated at the antenna elements. Similarly, some previous works that focused on vBS include [56] which proposed a theoretical model of CPU power consumption as a function of the active CPU cores, clock speed, and load. It also assumes a linear relation of traffic with computational load, and hence with the consumed power. This assumption is not universal, however, and our findings agree with previous studies finding *non-linear effects* [45].

More importantly, the impact of hardware, software platform, and context on these metrics is unknown and cannot be captured in predefined models. Our GP-based approach overcomes this obstacle since it essentially builds the models on-the-fly using the sampled data.

1. In femtocells, the BBU consumes 40% of power [55]



(a) (b)

Fig. 1: **(a)**: Comparison of power consumption at: the BBU (Intel NUC i7-8559U@2.70GHz), the BBU's CPU, and the RU (an USRP SDR), with 20Mbps DL and UL traffic. **(b)**: Consumed power over the baseline for different radio bandwidths and hardware platforms. SF PC 1: Intel NUC i7-8559U@2.70GHz; SF PC 2: Intel NUC i7-8650U@1.90GHz; Server 1: Dell XPS 8900 i7-6700@3.40GHz; Server 2: Dell Aurora R5 i7-9700@3.00GHz.

3 Preliminary Experimental Analysis

We performed experiments using a customized srsLTE-based testbed [2], described in Section 6.1. We present here results that motive the problem and our solution approach.

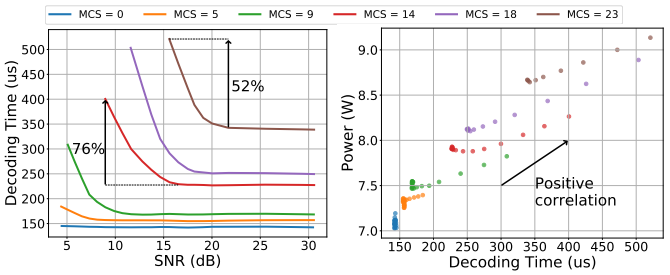
• BBU/CPU Power Cost & Impact of Platform. Our first finding is that the power consumption associated with the BBU processing is *comparable* to the RF chain's transmission power. This result is consistent with previous studies; for example, [55] estimated that 40% of a femtocell's power consumption is due to its BBU. In detail, Fig. 1a dissects the power consumption of a vBS deployed on a small factor (SF) PC, and presents the different power components stemming from the BBU's CPUs[2]; the BBUs cloud platform *except the CPUs*; and the actual radio unit (RU) which is deployed over an USRP software-defined radio. In order to have a complete picture, we measure the power consumption in four different scenarios: (i) the vBS is not deployed (baseline), (ii) the vBS is deployed with an idle user attached (vBS idle), (iii) the vBS is transmitting 20Mbps of downlink (DL) traffic, and (iv) the user is transmitting 20Mbps of uplink (UL) traffic to vBS.

Excluding the baseline scenario, the CPU power consumption is, on average, 29% larger than the RU power consumption; while the overall BBU power exceeds it by 175% (208% with full UL load). Interestingly, these numbers depend on the platform which hosts the BBU. Namely, Fig. 1b shows the BBU consumption over the baseline for various platforms.[3] We compare the power consumed by the BBU in idle state and when operating at full UL/DL buffer, and subtract the baseline power. Indeed, the power consumption changes significantly, and it is also affected by the vBS bandwidth – yet another configurable parameter of softwarized base stations.

• Impact of SNR & MCS. The second finding is that the signal-to-noise ratio (SNR) of the wireless channel and the UL modulation and coding scheme (MCS) affect the BBU

2. We use Intel's Running Average Power Limit function integrated into the Linux kernel for the CPU power consumption.

3. The small PCs consume less power than the servers, which can host more vBSs and thus consumes less power/user.



(a) (b)

Fig. 2: vBS over SF PC 1 at full UL buffer. **(a)**: UL decoding time for various SNR and MCS values. **(b)**: Power consumption as a function of the decoder performance (high correlation).

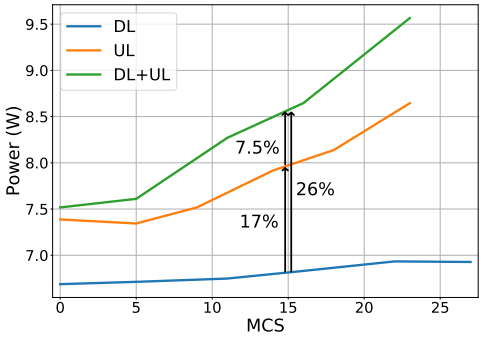
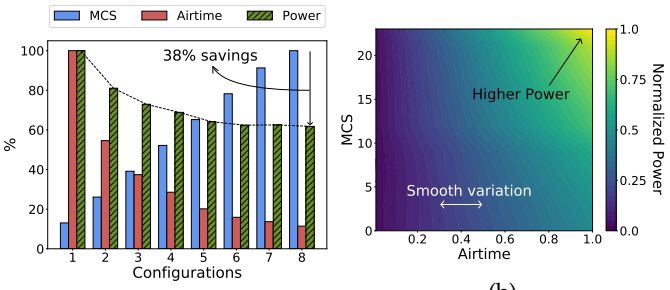


Fig. 4: MCS impact on BBU power consumption with high SNR. Results presented for three cases: only DL traffic being processed; only UL traffic; concurrent DL and UL traffic.



(a) (b)

Fig. 3: **(a)**: Eight configurations of MCS and airtime that offer 2.6Mbps in UL, and the respective power (idle mode's power subtracted). **(b)**: Normalized BBU power consumption over baseline, for full buffer UL transmissions and high SNR values, as a function of MCS and airtime.

computing load – and hence its power consumption – in a non-linear fashion. This is because the decoder needs increasingly more iterations when the received signal becomes noisier. Thus, the decoding time per subframe increases, e.g., by 52% between 20 and 15 dBs for MCS 23, see Fig. 2a; and this induces a commensurate increase in power consumption, see Fig. 2b. Besides, Fig. 2b shows that, even for a fixed decoding time, higher MCS values induce more power consumption, which is attributed to their more intricate demodulation (denser constellation map). Importantly, excessive decoding delays can induce throughput loss since they lead to violations of vBS processing deadlines [2]. Hence, maximizing throughput does not only have an unpredictable effect on power, but it is indeed highly non-trivial to achieve in a resource-efficient way.

• Configuration Options & Impact of Scheduler. The vBS orchestration difficulties are exacerbated by the plenitude of configuration options these base stations offer. Fig. 3a, for instance, presents combinations of MCS and airtime values (percentage of used subframes) achieving the same UL throughput. Configurations with higher MCSs (and therefore lower airtime) reduce power by 38%. However, this relation is *non-monotonic*, as we have also measured higher power when the MCS increases and SNR is relatively low. This latter effect is due to the fast increase of computing load (see Fig. 2b). On the other hand, configurations 6 to 8 have the same power consumption, but still differ since configuration 8 involves lower airtime and thus can serve more users, while configuration 6 is more resilient to noise. These decisions are made by the vBS radio scheduler[4] that selects the MCS and airtime based on the measured SNR (*context*). For this experiment, we have properly modified the srsLTE scheduler in order to support different airtime values. Fig. 3b shows the power consumption as a function of MCS and airtime for UL transmissions. We observe that both parameters have a smooth impact on power consumption, but it is important to stress that, in practice, this relation is not available and needs to be learned.

• Coupling of DL & UL Processing. Finally, Fig. 4 shows the BBU power consumption when DL and UL traffic is processed separately and concurrently (UL+DL), for high SNR and various MCS values. We observe that the joint power is not the total sum of the separate components. For instance, for MCS 15, concurrent DL and UL processing consumes just 7.5% more than UL-only processing (and 26% over DL-only). This is because there are common power consumption factors in both streams. This, in turn, makes it difficult to predict the overall vBS power consumption, given that the DL and UL can be configured separately. Also, note that UL power costs are higher and more volatile than DL, since decoding is more computationally demanding.

Conclusions: characterizing the vBS performance and power consumption is intricate as it depends on exogenous conditions such as the network traffic and SNR; and the BS configuration, e.g., the selected MCS and airtime parameters. There are many DL and UL configurations and some of them present *non-linear and non-monotonic* relations with power and throughput. Moreover, the power consumption depends on the BBU platform and the radio scheduler – which if almost fully customizable in vBSs. This hinders the derivation of generally applicable power consumption models. Hence, we propose the use of *online learning* to profile each vBS power cost and performance, and devise accordingly goal-driven configuration policies.

4. For example, our testbed's scheduler selects the maximum MCS for a given SNR and reduces the airtime whenever UL traffic is lower than the link capacity; but for DL traffic it selects lower MCSs so as to make the communication more robust, but this increases the power consumption.

4 System Model and Problem Formulation

Our modeling approach follows carefully the latest O-RAN architecture proposals [6] which have provisions for (in fact, envision) learning-based orchestration of the BS operation, and as such is fully aligned with the ideas presented in this work. We start by presenting the O-RAN elements that are pertinent to our model and subsequently we formulate the two optimization problems.

4.1 O-RAN Background and Model

We consider a virtualized Base Station (vBS) comprising a Baseband Unit (BBU) that may correspond to a 4G eNB or a 5G gNB[5] hosted in a cloud platform and attached to a Radio Unit (RU), which are fed by a (possibly) constrained energy source. This type of BSs is relevant for low-cost small cells, Power-over-Ethernet (PoE) cells, and other similar platforms that are increasingly common in 5G-and-Beyond networks. Our goal is to use O-RAN's control architecture to implement configuration policies that are adaptive to system dynamics while satisfying different energy-aware performance criteria.

O-RAN Architecture. Fig. 5 shows the high-level architecture of our system, which is O-RAN compliant [6]. The Learning Agent (LA) implements online learning algorithms within the Non-Real-Time (Non-RT) RAN Intelligent Controller (RIC) in the system's orchestrator, and selects efficient *radio policies* every orchestration period $t = 1, \dots, T$ (usually in the order of seconds). The optimal decision (i.e., a radio policy) in each t depends on the *context* information. This is provided at the beginning of each period by the vBS (via the O1 interface) from measurements collected at sub-second granularity within the near-RT RIC (using the E2 interface). The computed radio policies are then configured on the vBS via its A1-P interface as shown in Fig. 5. At the end of each orchestration period, the Data Monitor module in the Near-RT RIC computes a *reward* by aggregating the adopted performance metrics, which are collected from the vBS via the E2 interface; and eventually provides the results to the LA (O1 interface). Our system model and solution algorithms are fully compatible with this architecture.

Context Information. We define the DL context at each period t as $\omega_t^{dl} := [\bar{c}_t^{dl}, \tilde{c}_t^{dl}, d_t^{dl}]$, where $\bar{c}_t^{dl}$ and $\tilde{c}_t^{dl}$ are the mean and variance of the DL channel quality indicator (CQI) across all users in the previous period; and d_t^{dl} is the *new* bit arrivals at the vBS DL aggregated across all users. Note that the DL CQI values are sent periodically from the UEs to vBS through Uplink Control Information (UCI) carried by 4G/5G's Physical Uplink Shared Channel (PUSCH) or Physical Uplink Control Channel (PUCCH). Conversely, d_t^{dl} is measured by the vBS at the PDCP layer.

Also, we define the UL context as $\omega_t^{ul} := [\bar{c}_t^{ul}, \tilde{c}_t^{ul}, d_t^{ul}]$. The UL CQI is measured by the vBS at MAC layer, and the new UL bit arrivals are estimated from the periodic Buffer Status Reports (BSRs) of the users (UEs). All these measurement are collected by the Near-RT RIC's Data Monitor (Fig. 5) from the vBS using the E2 interface at sub-second granularity, and are aggregated at the start of each orchestration period t. We denote the global context vector $\omega_t := [\omega_t^{dl}, \omega_t^{ul}] \in \Omega$, where Ω is the context space. Note that the contexts are related to the traffic load and channel quality and are exogenous parameters, i.e., the configuration decisions cannot affect them. This allows us to formulate the problem as a Contextual Multi-armed Bandit or Contextual Bandit (CB). By using this formulation we can configure the system based on the observed contexts and learn from the zeroth-order feedback of our system (i.e., we observe only the outcome of the employed configuration).

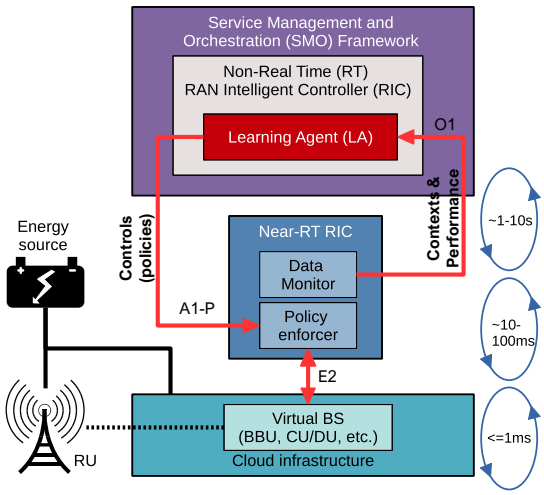


Fig. 5: O-RAN compliant system architecture and workflow.

vBS Controls. We define the DL control $x_t^{dl} := [p_t^{dl}, m_t^{dl}, a_t^{dl}]$ at period t, where $p_t^{dl} \in \mathcal{P}^{dl}$ is a *transmission power control (TPC) policy* for the maximum allowed vBS transmission power, $m_t^{dl} \in \mathcal{M}^{dl}$ is the highest MCS eligible by the vBS (*DL MCS policy*), and $a_t^{dl} \in \mathcal{A}^{dl}$ is the maximum vBS transmission airtime (*DL airtime policy*). We define the UL control $x_t^{ul} := [m_t^{ul}, a_t^{ul}]$, where $m_t^{ul} \in \mathcal{M}^{ul}$ and $a_t^{ul} \in \mathcal{A}^{ul}$ are the UL MCS and airtime policies.[6] We hence formalize each control at decision period t as a *radio policy*:

$$x_t := [x_t^{dl}, x_t^{ul}] \in \mathcal{X}, \qquad \mathcal{X} = \mathcal{P}^{dl} \times \mathcal{M}^{dl} \times \mathcal{A}^{dl} \times \mathcal{M}^{ul} \times \mathcal{A}^{ul},$$

where $\mathcal{X}$ is the control space. Once computed, the LA sends each radio control policy to the Near-RT RIC via O-RAN's A1-P interface, which is then applied to vBS. The UL policies are applied by configuring each UL scheduling at the vBS MAC layer.

Rewards. We denote with $R^{dl}(\omega_t^{dl}, x_t^{dl})$ and $R^{ul}(\omega_t^{ul}, x_t^{ul})$ the DL and UL data transmission rates, and define the *reward* function $r(\omega_t, x_t) :=$

$$\log\left(1 + \frac{R^{dl}(\omega_t^{dl}, x_t^{dl})}{d_t^{dl}}\right) + \log\left(1 + \frac{R^{ul}(\omega_t^{ul}, x_t^{ul})}{d_t^{ul}}\right) \quad (2)$$

where the logarithms are used to achieve fairness between the DL and UL flows – and to that end, one could use any other α-fair function [57]. Note that we divide the achieved rates with the actual load in the respective stream (uplink or downlink) since the reward should naturally be defined in relation to the needs of the system. Also, it is

5. 5G decouples BBU in 2 logical functions, i.e., a central unit (CU) and a distributed unit (DU). Our scheme controls the DU, or both when these are co-located.

6. We do not define an UL TPC policy since the users' transmission power has less impact on the vBS power than the MCS and UL airtime; but our framework can be readily extended to include this decision.

important to stress that in practice we can only hope to observe **noisy values** of these functions, even when their arguments are fixed, because naturally the system operation is stochastic and also the power measurements are noisy – as we have indeed seen in our experiments. Fortunately, our optimization framework can handle such impairments. Henceforth, we denote with $R_t^{dl}(\omega_t^{dl}, x_t^{dl})$, $R_t^{ul}(\omega_t^{ul}, x_t^{ul})$ and $r_t(\omega_t, x_t)$ these noisy samples of the functions at period t, which are considered to be stationary and return the mean (unperturbed) respective values when averaged (i.e., on expectation).

4.2 Case 1: Balancing performance and cost

We start with the case where the power supply is scarce or, equivalently, the operator wishes to reduce the power consumption costs. This can be achieved with a scalarized *objective function*:

$$u(\omega_t, x_t) := r(\omega_t, x_t) - \delta \cdot B\big(P(\omega_t, x_t)\big), \tag{3}$$

where $P(\omega_t, x_t)$ is the vBS power consumption associated with the pair context-control (ω_t, x_t), $B(\cdot)$ is a smooth function that models the cost associated with power consumption, and parameter δ determines the relative importance of the power cost and achieved throughput, and can be selected based on the operator's preferences. We will also use $u_t(\omega_t, x_t)$ to denote the realization of the objective function related to the t-period samples $P_t(\omega_t, x_t)$ and $r_t(\omega_t, x_t)$. The selection of the cost function is crucial here. In the simplest case, it can be a linear function that maps the actual consumed power to a monetary value (negative reward). But, it can also model situations where policies that exceed a power threshold should be prevented due to regulation, battery constraints, and so on. To capture all these cases, we propose to use a parameterized sigmoid function with sharpness and tipping parameters a and b:

$$B(x) := \frac{1 + e^{ab}}{e^{ab}} \left(\frac{1}{1 + e^{-a(x-b)}} - \frac{1}{1 + e^{ab}} \right). \tag{4}$$

When $a \to 0$, function $B(\cdot)$ approximates a linear function, and when a grows [58] it approximates the step function, without however to induce unbounded gradients – a condition that would deteriorate the learning process.

Following the standard approach in Bayesian bandit optimization [13], [39], we use the cumulative contextual regret to assess the performance of our algorithm. Namely, we define the average T-period contextual regret:

$$R_T := \sum_{t=1}^{T} \left(\max_{x' \in \mathcal{X}} u(\omega_t, x') - u(\omega_t, x_t) \right),$$

where $\max_{x' \in \mathcal{X}} u(\omega_t, x')$ yields the best decision for the current period, which we cannot calculate in practice since the objective function is unknown. Our goal, therefore, is to find a sequence of decisions $\langle x_t \rangle_{t=1}^{T}$ from set $\mathcal{X}$ which ensure asymptotically sublinear average pseudo-regret, i.e., $\lim_{T \to \infty} E[R_T]/T = 0$, where the expectation is taken with respect to the noisy samples and the context arrival process.

4.3 Case 2: Hard power budget

A different problem arises when the vBS operates under a hard power budget $P_{\max}$, e.g., when powered over Ethernet. In these cases, the LA has to find the maximum-throughput configuration that respects the available power budget. Importantly, the LA needs to achieve this goal by emloying a *safe* exploration of the configuration space $\mathcal{X}$ in order to satisfy the $P_{\max}$ threshold at any period, i.e., not only at the final optimal-operation stage. We define the respective regret:

$$R_T^s := \sum_{t=1}^{T} \left(\max_{x' \in S_t(\omega_t)} r(\omega_t, x') - r(\omega_t, x_t) \right), \tag{5}$$

where in this case the decisions are selected from set

$$S_t(\omega_t) = \Big\{ x \in \mathcal{X} \;\Big|\; P(\omega_t, x) \leq P_{\max} \Big\}. \tag{6}$$

Note that we use in the definition of regret directly the throughput reward, since the power is now considered a hard constraint. Our goal is to find a sequence $\langle x_t \rangle_{t=1}^{T}$, $x_t \in S_t(\omega_t)$, such that $\lim_{T \to \infty} E[R_T^s]/T = 0$. It is important to stress that the sets $S_t(\omega_t), \forall \omega_t$, are unknown initially, since $P(\omega, x)$ is also unknown, and therefore we need learn them using the real-time measurements $P_t(\omega_t, x_t)$. Similarly, we only have access to r_t and u_t, i.e., the t-period noisy measurements, instead of the actual functions r and u.

To solve the above problems, we propose a non-parametric learning approach using Gaussian Processes, Contextual Bandits, and Bayesian learning. Our approach has the additional practical advantage that one can change $P_{\max}$ in runtime, which in fact is possible in the PoE standard (IEEE 802.3bt), at any time without having to restart the learning process. Other parametric methods, such as Reinforcement Learning relying on neural networks, need to be re-trained if the constraint changes, which naturally increases substantially the required training data.

5 Bayesian Online Learning solutions

Next, we propose two online algorithms for solving the problems stated in Sections 4.2 and 4.3. Our proposals leverage state-of-the-art Bayesian learning techniques which are properly configured and extended to account for the network context information, and amended with practical rules (of independent interest) that improve their performance, as we verify experimentally.

5.1 BP-vRAN: Balancing performance and cost

Many algorithms for solving contextual bandit problems assume there is a feature vector associated with each action, and the objective function is linear in that vector [59], [60]. This assumption does not hold here for the following reasons. Firstly, the objective function is not linear, see eqs. (2)-(4). Secondly, the function values associated with different actions (i.e., vBS control policies) are correlated. Intuitively, we can think that a small change in some parameter (e.g., airtime) will induce a small change in the vBS consumed power. This is actually evaluated experimentally in Fig. 3b. This means that we can obtain information about unobserved context-control pairs by observing nearby actions, thus reducing the exploration time.

Based on these observations, we propose a Bayesian optimization method where we model the objective function as a sample from a Gaussian Process (GP) over the joint context-control space. This non-parametric estimator captures the aforementioned non-linearities and correlations, and provides predictive uncertainty on the function estimation. Hence, enable us to address effectively the exploration - exploitation trade-off.

Function estimator. We use a GP as a function estimator, which is a collection of random variables following joint Gaussian distributions [11]. Let $z \in \mathcal{Z} = \Omega \times \mathcal{X}$ denote a context-control pair. We model the unknown objective function (3) as a sample from a $GP(\mu(z), k(z, z'))$, where $\mu(z)$ is its mean function and $k(z, z')$ is its covariance function or kernel. Without loss of generality, we assume $\mu = 0$ and bounded variance $k(z, z) < 1$, which we refer to as the *prior distribution*, not conditioned on data.

Given this prior and a set of observations, the mean and covariance of the *posterior distribution* can be computed using closed form formulas. Let $y_T = [u_1, \ldots, u_T]$ be a vector of noisy samples (assuming *i.i.d.* Gaussian noise $\sim N(0, \zeta^2)$) at points $Z_T = [z_1, \ldots, z_T]$. Then, the posterior distribution of the objective function follows a GP distribution with mean $\mu_T(z)$ and covariance $k_T(z, z')$:

$$\mu_T(z) = k_T(z)^\top (K_T + \zeta^2 \mathbf{1}_T)^{-1} y_T \tag{7}$$

$$k_T(z, z') = k(z, z') - k_T(z)^\top (K_T + \zeta^2 \mathbf{1}_T)^{-1} k_T(z') \tag{8}$$

where $k_T(z) = [k(z_1, z), \ldots, k(z_T, z)]^\top$, $K_T(z)$ is the kernel matrix $[k(z, z')]_{z,z' \in Z_T}$, and $\mathbf{1}_T$ is the T-dimension identity matrix. These equations allow us to estimate the distribution of unobserved values of z based on the prior distribution, the vector Z_T, and the function observations y_T.

Kernel function. The kernel selection is crucial as it shapes the prior and posterior GP distributions by encoding the correlation between the values of the objective function of every pair of points. Namely, $k(z, z')$ indicates the similarity between $u_t(z)$ and $u_t(z')$. In other words, the kernel characterizes the smoothness of the function [61]. The properties of the kernel function should be carefully selected according to the specific application and the underlying function that will be learned. Therefore, we use the experimental data analyzed in Sec. 3 to conclude that our kernel should satisfy two properties: *stationarity* and *anisotropicity*. On the one hand, the kernel $k(z, z')$ is stationary since it depends only on the distance of z from z', which means it is invariant to translations in $\mathcal{Z}$. On the other hand, a kernel is anisotropic since the encoded smoothness is different among the different dimensions of $\mathcal{Z}$. That is, the kernel is not invariant to rotations in $\mathcal{Z}$. The smoothness of the different dimensions of the function u are encoded into a length-scale vector $\mathcal{L} = [l_1, \ldots, l_N]$, where N indicates the number of dimensions of $\mathcal{Z}$. Thus, the distance between two points based on the length-scale vector can be written as:

$$d(z, z') = \sqrt{(z - z')^\top L^{-2} (z - z')}, \tag{9}$$

where $L = \text{diag}(\mathcal{L})$ is a diagonal matrix of the length-scale values. There are several kernel functions satisfying these properties such as the squared exponential kernel, one of the most commonly used. However, this kernel function assumes the underlying function to be very smooth, i.e., infinitely differentiable. This assumption does not hold in our framework since function $B(\cdot)$ defined in eq. (4) is not infinitely differentiable. Besides, recall that $B(\cdot)$ maps the monetary cost associated with the consumed power and can be defined according to the operator's needs. For that reason, we relax this assumption and select the anisotropic version of the Matérn kernel, which also satisfies the properties discussed above [11]. Furthermore, we configure it with parameter $\nu = \frac{3}{2}$, which implies that the objective function is at least once differentiable. Note that this is a mild assumption, which yields a loose regret bound (see Lemma 1). In fact, our experimental evaluation in Section 3 shows that our approach performs much better than our theoretical bounds in the scenarios we tested. However, if we had more information about the structure of the function to learn, we could easily tighten such bound by selecting higher values of ν or by using a squared exponential kernel, which may improve the rate of increase of information gain. In this paper, we opt for the most conservative choice to cover scenarios beyond the ones shown in our experimental evaluation. The expression of the selected kernel is given by:

$$k(z, z') = (1 + \sqrt{3} d(z, z')) \exp(-\sqrt{3} d(z, z')). \tag{10}$$

To improve performance, we can optimize the hyperparameters $\mathcal{L}$ and the noise variance ζ^2, eq. (7)-(8), before running the algorithm, by maximizing the likelihood estimation over prior data and keep these values constant over time. A different approach, namely when the hyperparameters are optimized using the data acquired in runtime, it is not guaranteed that the GP's confidence interval will cover the true function, and hence might induce the optimization process to stuck in poor local optima [62]. We have also observed this in our experiments.

Acquisition function. The acquisition function selects one control x_t at each period t based on the posterior distribution of the objective function over the context-control pairs. To this aim, we use the Upper Confidence Bound (UCB) method which follows the principle of *optimism in the face of uncertainty* and allows us to derive theoretical guarantees for the algorithm. Formally:

$$x_t = \underset{x \in \mathcal{X}}{\text{argmax}} \ \mu_{t-1}(\omega_t, x) + \sqrt{\beta_t} \sigma_{t-1}(\omega_t, x). \tag{11}$$

where ω_t is the observed context at time t, β_t is a weighting parameter and $\sigma_t^2(z) = k_t(z, z)$. We formalize our approach, which we refer to as BP-vRAN (Bayesian optimization for Power consumption in vRANs), in Algorithm 1. At the beginning of each decision period t a context ω_t is observed (line 4). Based on the observed context ω_t and the vectors Z_{t-1} and y_{t-1}, the posterior distribution is computed using eqs. (7) and (8) (line 5). Note that when we have no data ($y_0 = \emptyset$, $Z_0 = \emptyset$) the posterior distribution is equal to the prior distribution. The control x_t is decided based on the GP posterior and the acquisition function (line 6). At the end of t, the throughput and consumed power are observed (line 7). Then, the reward and the monetary cost of the power are computed using eqs. (2) and (4), respectively. With these values, the value of the objective is computed using eq. (3) (line 8). Finally, the new context-control pair z_t and the value of the objective function $u_t(\omega_t, x_t)$ are

Algorithm 1 BP-vRAN: Performance and cost balancing

1: **Inputs:** Control Space $\mathcal{X}$, kernel k, β
2: **Initialize:** $y_0 = \emptyset$, $Z_0 = \emptyset$
3: **for** $t = 1, 2, \dots$ **do**
4: Observe the context ω_t
5: Compute μ_{t-1} and $\sigma^2_{t-1} = k_{t-1}(z_t, z_t)$, eqs. (7)-(8)
6: $x_t = \text{argmax}_{x \in \mathcal{X}} \ \mu_{t-1}(\omega_t, x) + \sqrt{\beta_t}\sigma_{t-1}(\omega_t, x)$
7: Measure $R^{dl}_t(\omega^{dl}_t, x^{dl}_t)$, $R^{ul}_t(\omega^{ul}_t, x^{ul}_t)$ and $P_t(\omega_t, x_t)$ at the end of the decision period t
8: Compute $u_t(\omega_t, x_t)$ using (2), (3) and (4)
9: Update $Z_t \leftarrow Z_{t-1} \cup z_t := [\omega_t, x_t]$
10: Update $y_t \leftarrow y_{t-1} \cup u_t(\omega_t, x_t)$
11: **end for**

included in the vectors Z_t and y_t, respectively, to improve the posterior distribution of the next iteration (lines 9-10).

Note that an alternative formulation of BP-vRAN with two GPs (to approximate the reward and the consumed power separately) instead of one is amenable to better optimization of the kernels' hyperparameters. Nevertheless, the posterior variance of the objective function can be arbitrarily hard to obtain since the monetary cost of the power ($B(\cdot)$) is selected by the operator according to its needs. In addition, this approach doubles the computational and memory requirements.

Theoretical results. The choice of a value for β_t in eq. (11) is very important since it controls the trade-off between exploration and exploitation. Larger values of β_t lead the acquisition function to select controls with higher uncertainty while, conversely, controls already known to be high-performing (though not necessarily *highest-performing*) are selected when β_t takes smaller values. Following [39], we select

$$\beta_t = 2B^2 + 300\gamma_t \ln^3(t/\epsilon) \tag{12}$$

where $\epsilon \in (0, 1)$, $B \geq \|u\|_k$ is an upper bound on the Reproductive Kernel Hilbert Space (RKHS) norm of u, and γ_t is the maximum mutual information gain obtained from u after t observations have been collected.

Lemma 1. *The contextual regret R_T of BP-vRAN satisfies*

$$P\left(R_T \leq \sqrt{C_1 T \beta_T \gamma_T} \ \forall T \geq 1\right) \geq 1 - \epsilon, \tag{13}$$

at stage T, where $C_1 = \frac{8}{\log(1+\zeta^{-2})}$ and $\gamma_t = \mathcal{O}(t^{44/45}\log(t))$.

The proof of Lemma 1 is given in the Appendix. For the derivation of the bound of the information gain γ_t, we consider a Matérn kernel with $\nu = \frac{3}{2}$ and $N = 11$ dimensions in $\mathcal{Z}$, which correspond to a 6- and a 5-dimensional context and control space, respectively, as described in Sec. 4. For this setting, we particularize the expression provided in Theorem 5 of [63] to obtain the bound $\gamma_t = \mathcal{O}(t^{44/45}\log(t))$. Note that the regret bound obtained in this analysis considers a worst-case scenario, while the performance of the algorithm in practice is commonly far from these bounds as shown in Sec. 6. It is worth mentioning, however, that the bound provided in Lemma 1 indicates that BP-vRAN is a no-regret algorithm, i.e., $\lim_{T\to\infty} E[R_T]/T = 0$.

5.2 SBP-vRAN: Safe Bayesian Optimization

Imposing hard constraints as proposed in Sec. 4.3, compounds the problem. Prior works, e.g., in robotics and other areas [12], [13], [64], [65], have proposed Bayesian optimization algorithms with *safety constraints*. Their main idea lays upon the definition: every t we define a subset of *safe* controls $S_t \subseteq \mathcal{X}$ that satisfy the constraints with certainty. Then, it is needed to interleave an exploration process so as to expand the safe set, while seeking a safe action with high performance. Unfortunately, these works do not consider contextual information, which clearly affects the safe set, i.e., $S_t(\omega_t) \subseteq \mathcal{X}$. To the best of our knowledge, only SafeOpt [65] proposes a contextual safe learning algorithm. However, although that algorithm provides theoretical guarantees, its acquisition function selects the control with the highest uncertainty among all candidates that can expand the safe set and also the potential maximizers. We found in our experiments that this approach has overly slow convergence. This practical issue has been reported in other works as well, e.g. [66]. Hence, we improve this methodology by employing the acquisition function of CGP-UCB [39], but *constrained to the safe set.*

We denote $y^f_T = [r_1, \dots, r_T]$ the vector of reward samples at T and $y^c_T = [P_1, \dots, P_T]$ the power consumption samples. We use one GP for the reward and one for the power constraint. Both GPs have the same prior distribution and kernel but different hyperparameters. The posterior distribution can be computed using (7)-(8), and replacing y_T by y^f_T or y^c_T, for each GP. We denote the posterior mean and covariance of the reward at T as $\mu^f_T(z)$ and $k^f_T(z, z')$, and $\mu^c_T(z)$ and $k^c_T(z, z')$ for the power, respectively. The initial safe set $S_0 \subseteq \mathcal{X}$ is common for all contexts, and includes low power consumption configurations (vBS close to idle). This is worst-case S_0 can be expanded using prior data.

At each period, S_t is computed based on the posterior distribution of the power consumption provided by the GP. We assume the true value of the power consumption at time t is within the interval $[\mu^c_t(z) \pm \beta_t\sigma^c_t(z)]$, where $\sigma^c_t(z) = k^c_t(z, z)$. Using the posterior distribution, we define the safe set a time t and for a given context ω_t as:

$$S_t = \left\{x \in \mathcal{X} \ \middle| \ \mu^c_{t-1}(\omega_t, x) + \beta_t\sigma^c_{t-1}(\omega_t, x) \leq P_{\max}\right\}. \tag{14}$$

The controls are selected at each period t using the CGP-UCB policy subject to the safe set:

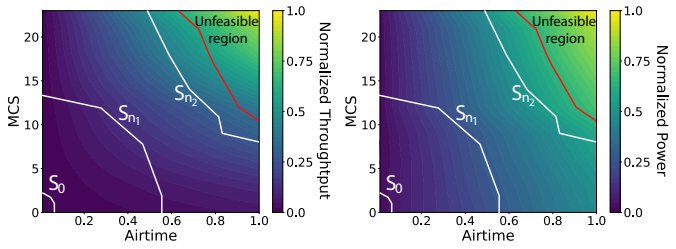
$$x_t = \underset{x \in S_t}{\text{argmax}} \ \mu^f_{t-1}(\omega_t, x) + \sqrt{\beta_t}\sigma^f_{t-1}(\omega_t, x), \tag{15}$$

where $\left(\sigma^f_t(z)\right)^2 = k^f_t(z, z)$.

We summarize our approach, named SBP-vRAN (Safe Bayesian optimization for Power consumption in vRANs), in Algorithm 2. It is worth mentioning that in many practical scenarios it is desirable to have a soft constraint instead of a hard constraint. For instance, we may be interested in violating the soft constraint (increase the power consumption) to avoid poor user performance. We provide two alternatives to handle this scenario. First, we can use BP-vRAN by designing $B(\cdot)$ such that a power consumption exceeding the constraint incurs in high monetary cost. This approach provides soft guarantees where the power

Algorithm 2 SBP-vRAN: Safe online optimization

Inputs: Control Space $\mathcal{X}$, Initial safe set S_0, kernel k, β, $P_{\max}$
Initialize: $y_0^f = \emptyset$, $y_0^c = \emptyset$, $Z_0 = \emptyset$
for $t = 1, 2, \ldots$ **do**
  Observe the context ω_t
  Compute μ_{t-1}^f, σ_{t-1}^f, μ_{t-1}^c and σ_{t-1}^c using eqs. (7)-(8)
  $S_t = S_0 \cup \{x \in \mathcal{X} \mid \mu_{t-1}^c(\omega_t, x) + \beta_t \sigma_{t-1}^c(\omega_t, x) \leq P_{\max}\}$
  $x_t = \text{argmax}_{x \in S_t} \ \mu_{t-1}^f(\omega_t, x) + \sqrt{\beta_t}\sigma_{t-1}^f(\omega_t, x)$
  Measure $R_t^{dl}(\omega_t^{dl}, x_t^{dl})$, $R_t^{ul}(\omega_t^{ul}, x_t^{ul})$ and $P_t(\omega_t, x_t)$ at the end of the decision period t
  Compute $r_t(\omega_t, x_t)$ using (2)
  Update $Z_t \leftarrow Z_{t-1} \cup [\omega_t, x_t]$
  Update $y_t^f \leftarrow y_{t-1}^f \cup r_t(\omega_t, x_t)$
  Update $y_t^c \leftarrow y_{t-1}^c \cup P_t(\omega_t, x_t)$
end for



(a) (b)

Fig. 6: Example of safe set expansion in the uplink throughput and power domains as a function of two decision variables: uplink MCS and uplink airtime. As SBP-vRAN explores, the initial safe set S_0 is expanded until it reaches the boundary of the unfeasible region, where the optimum is located.

constraint will be met in average but not at every interval. Alternatively, we can modify the definition of the safe set in eq. (14). Thus, we can add an exception such that if the expected performance of all actions in the safe set is below a performance threshold $r_{\min}$, include at least one action whose expected performance is higher than $r_{\min}$. Using this mechanism, we can set a minimum performance requirement for the vBS operation.

Convergence of SBP-vRAN. Note that SBP-vRAN does not expand explicitly the safe set, like in other works such as [13], [65]. In general, an explicit expansion of the safe set is needed (e.g., by exploring the controls in the boundary) to converge to the true safe set and therefore to reach the optimal safe control. However, we found that our acquisition function can both maximize the performance and expand the safe set at the same time under some conditions.

Let us assume that the objective function and the constrained function are smooth and positively correlated. In this case, the maximization of the objective function also implies the expansion of the safe set. In fact, the optimal configuration is located at the boundary of the constraint space. This is a reasonable assumption in practice, as we can assess empirically: On the one hand, Fig. 6a shows the uplink throughput of our vBS as a function of the MCS and the airtime (two of our control actions). From this figure, we can see that the higher the MCS and the airtime the higher the throughput. On the other hand, Fig. 6b shows the consumed power as a function of the same variables. Note that both figures show the same trend: the higher the throughput the higher the consumed power.

We should remark that we have only considered two vBS controls (MCS and airtime) for this example. However, although the power behavior becomes non-linear when including all the dimensions of the problem, these conclusions also hold in the complete problem. It is obvious that higher airtime provides higher throughput. It is also evident that higher MCSs provide higher throughput under feasible conditions (appropriate SNR) as they allow to pack more data symbols per unit of time. Similarly, higher MCSs incur in higher power consumption because the number of computations required by the decoding algorithms scale linearly with the number of bits to decode. Moreover, higher transmission power enables higher MCSs and therefore higher throughput. Therefore, *higher throughput is generally associated with higher power consumption*.

The annotations in Figs. 6a-6b exemplify how SBP-vRAN expands the safe set. The initial safe set (S_0) is a set of configurations with the lowest power consumption, i.e., low MCS and airtime. This conservative initial safe set avoids violating the constraint from the beginning but also increases the convergence time. The aim of SBP-vRAN is to maximize the the reward function r which is directly related to the throughput. Moreover, our acquisition function in eq. (15) will select controls with high performance but also with high uncertainty. These conditions are met by the controls in the boundary of the safe set. By exploring these controls we are reducing the uncertainty of its neighborhood and therefore expanding the safe set. After a few iterations ($t = n_1$), the safe set S_{n_1} has been expanded and the algorithm can now select configurations with higher throughput. At that point, the algorithm will continue exploring the boundary of the constraint since it contains the configurations with the highest throughput and also high uncertainty. After a few iterations more, the safe set will reach the boundary of the constraint, finalizing its expansion: the optimal configurations fall into the boundary of the constraint space. This is demonstrated in the following experimental evaluation.

6 Experimental Evaluation

We have built a customized testbed to perform a thorough evaluation of the proposed ML resource orchestration techniques under realistic conditions. Our experiments employ the software-based eNB srsRAN, cf. [2], which we have properly modified (e.g., implementing scheduling policies, enabling airtime selection, etc.) so as to capture the entire range of our controls. The testbed configuration and created datasets are available online[7] for reproducibility reasons and, importantly, so as to facilitate further research in the area of AI/ML-assisted RAN orchestration.

7. https://github.com/jaayala/power_dlul_dataset

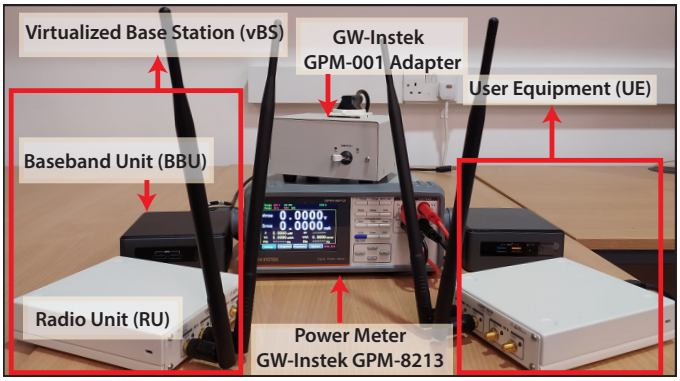


Fig. 7: Customized Wireless Testbed with a vBS (srsRAN) and a node aggregating the UE traffic. Measurements are collected in real time, using a GW-Instek Power Meter and a Power Adapter.

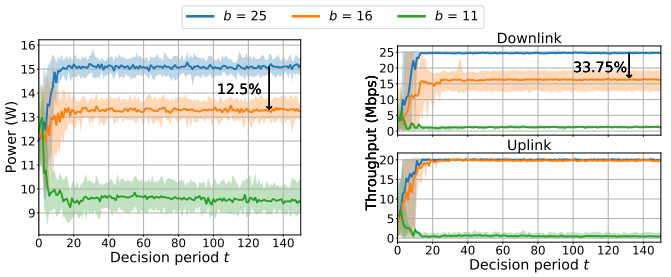


Fig. 8: Convergence rate evaluation of BP-vRAN for different objective function parameters.

6.1 Experimental setup

The testbed, shown in Fig. 7, comprises a vBS, the user equipment (UE)[8], and a digital power meter. Both the vBS and UE consist of an Ettus Research USRP B210 as RU, srseNB/srsUE (from srsRAN suite [2]) as BBU for the eNB and UE, and two small factor general-purpose PCs (Intel NUCs with CPU i7-8559U@2.70GHz) deploying each respective BBU and the near-RT RIC of Fig. 5. The vBS and UE are connected using SMA cables with 20dB attenuators and we adjust the gain of the RU's RF chains to attain different SNR values. Without loss of generality, we select a 10-MHz band that renders a maximum capacity of roughly 32 and 23 Mbps in DL and UL, respectively. We use the power meter GW-Instek GPM-8213 to measure the power consumption of BBU and RU by plugging their power supply cable to a GW-Instek Measuring adapter GPM-001. Finally, we have integrated E2's interface and the ability to enforce control policies *on-the-fly* (see Section 4) in srseNB.

We use three auxiliary PCs (not shown in the figure) hosting the non-RT RIC and the network traffic end hosts, which use *mgen*[9]. Finally, we have implemented O1 interface (Fig. 5) using the USB-based power meter SCPI (Standard Commands for Programmable Instruments) interface concerning power consumption measurements and a REST interface for the remainder. A final remark is that our RU (USRP B210) does not integrate a variable power amplifier. Instead, it uses a fixed power amplifier consuming 3W and a variable attenuator for power calibration (see Fig. 1a). To compensate for this, we post-process the power measurements to include a variable RU consumption according to a linear model based on previous works [48], [50] and a 3W cap.

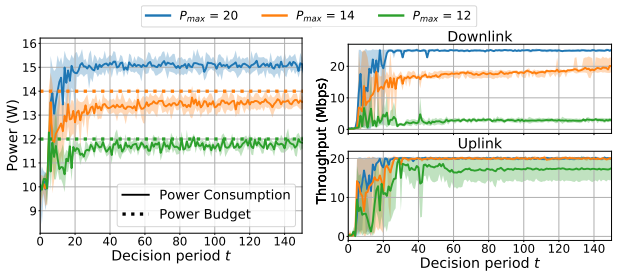
For the elaboration of the dataset used in Sec. 3, we configure the vBS and UE in order to fix the conditions in the uplink and the downlink in terms of traffic load, channel quality, MCS, and airtime. Then, we fix each configuration for approximately one minute while the system takes measurements that later are processed to obtain its statistics. We assess the power behavior of the vBS by measuring the power consumption of its CPU and the whole BBU, the achieved performance in terms of throughput and goodput, details about the decoder at the vBS such as the subframe decoding time and the number of turbo decoder iterations per subframe, and some MAC and PHY indicators such as the Buffer Status Report (BSR), Block Error Rate (BSR), and the used MCS and airtime. Moreover, we detect and identify unfeasible configurations in the dataset. This mainly occurs when an MCS value is forced but the channel quality is not good enough to decode its data. Finally, we release our dataset[7] online allowing the community to realistically emulate the behavior of a vBS in terms of power consumption and performance as a function of its configuration and conditions (user traffic load and channel qualities) for future research.

For the evaluation we consider $|\mathcal{P}^{dl}| = 20$, $|\mathcal{M}^{dl}| = 28$, $|\mathcal{M}^{ul}| = 24$, and $|\mathcal{A}^{dl}| = |\mathcal{A}^{ul}| = 11$, and therefore the size of the control set is $|\mathcal{X}| \approx 1.6 \cdot 10^6$. Note that, for a decision period of 10 seconds, we would need up to 185 days to explore every control policy in $\mathcal{X}$ once, which highlights the need for a data-efficient learning strategy. Although Lemma 1 guarantees convergence and sublinear regret in general, faster convergence can be achieved with problem-specific information. Hence, and in line with previous works [65], [66], we select $\beta^{1/2} = 2.5$, which shows good performance in our setup. In the case of BP-vRAN, we configure $\delta = 20$ and set the parameters a and b in the penalty function, eq. (4), to severely penalize the power consumption values close to b or higher. Namely, we set $a = 2.5$ and evaluate different values of b. Finally, we present the results of 10 (at least) experiments, where we plot the mean values and the 10^{th} and 90^{th} percentiles (shadowed areas). The source code of the algorithms BP-vRAN[10] and SBP-vRAN[11] used for this evaluation can be found online.

6.2 Convergence Evaluation

We start off by evaluating the convergence of BP-vRAN and SBP-vRAN. To this end, we consider the special case of a sin-

8. We use one UE emulating the load of multiple users (see in Sec. 6.3).
9. https://www.nrl.navy.mil/itd/ncs/products/mgen.
10. https://github.com/jaayala/contextual_bayesian_optimization
11. https://github.com/jaayala/constrained_bayes_opt



(a) (b)

Fig. 9: Convergence rate evaluation of SBP-vRAN for different values of the power budget $P_{\max}$.

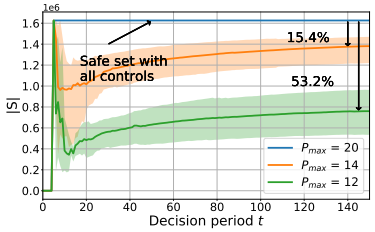


Fig. 10: Time evolution of safe set size of SBP-vRAN for different power budgets $P_{\max}$.

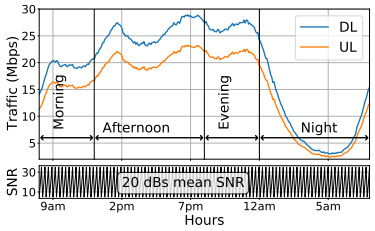


Fig. 11: One-day traffic pattern (top) and worst case channel quality pattern (bottom).

gle context and observe their performance over time *with no prior training up till they converge to optimal policies*. We select a context with high SNR = 35 dB (CQI = 15) in DL and UL, and high traffic demands (relative to our testbed's capacity) equal to 25 and 20 Mbps for DL and UL, respectively. Fig. 8-9 show the temporal evolution of different metrics for both algorithms during 150 orchestration periods.

Let us discuss first the results of BP-vRAN in Fig. 8. We observe that the power consumption and, consequently, throughput, are reduced for lower values of b, e.g., there is 12.5% power drop and 33.75% throughput drop between $b = 25$ and $b = 16$. This is intuitive because lowering b induces more stringent power requirements. Note that $b = 16$ only penalizes DL throughput. This is because it imposes a mild power requirement, and hence BP-vRAN *only* sacrifices transmission power, which reduces DL SNR and thus DL throughput. Lower values of b force BP-vRAN to sacrifice UL throughput too.

Concerning SBP-vRAN, we evaluate different values of $P_{\max}$ up to $P_{\max} = 20$, which is an upper bound for the power consumption irrespective of the policy and the context. The results, in Fig. 9, depict how SBP-vRAN learns to use configurations within the power budget *with high probability*, sacrificing throughput when so required. Note that, in all the cases, SBP-vRAN always selects policies very close to $P_{\max}$. This is because the optimal policy, i.e., the one that maximizes throughput, usually requires consuming all the $P_{\max}$ budget. To this end, SBP-vRAN gradually expands its safe set close to $P_{\max}$ and therefore an explicit strategy to expand the safe set is not needed. Specifically, Fig. 10 shows that all the controls are safe for $P_{\max} = 20$, with 15.4% and 53.2% less safe policies for $P_{\max} = 14$ and $P_{\max} = 12$, respectively. As expected, lower values of $P_{\max}$ incur a smaller safe policy set.

We conclude this evaluation with the observation that, despite using a large set of policies $\mathcal{X}$, both algorithms converge within 30 orchestration periods. This highlights the *data-efficiency* of our solutions, which discern optimal policies by observing only a small subset of $\mathcal{X}$.

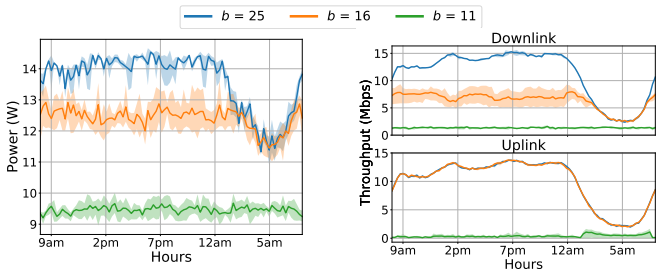
6.3 Performance in real network contexts

Next, we evaluate the performance of BP-vRAN and SBP-vRAN using a realistic one-day traffic pattern from [67] (Fig. 11, top). Concerning channel quality, we consider a worst-case pattern emulating UEs with high mobility (Fig. 11, bottom), which compromises network capacity (well below the demand). Due to the granularity of our traffic dataset, we set the orchestration period length to 5 minutes in these experiments (note there is no loss in generality). We run our algorithms for two days and present results of the second day to focus on the attained system performance. Their convergence, evaluated in the previous subsection, takes just a few periods. This is possible because the selected policies for correlated contexts are also correlated, i.e., knowledge acquired for one context is *transferred to other similar contexts*. Hence, after few iterations, the algorithms select efficient policies even for *unseen* contexts.

To remove the clutter introduced by the high SNR variability under evaluation, each point in Figs. 12 and 13 corresponds to the average across all the points of a SNR cycle, see Fig. 11, bottom. Fig. 12 shows the total power consumption (a) and the evolution of throughput along the day (b) using BP-vRAN and different configurations of the objective function. We observe that the power consumption evolves with the traffic demand and with the selected value of b. For instance, when $b = 16$, the achieved throughput is penalized in favor of better power consumption during daylight but no performance degradation is required during the night (between 2am and 7am). Similarly, Fig. 13 shows the performance of SBP-vRAN under the same scenarios. Specifically, SBP-vRAN manages to satisfy the power budget constraint with probabilities 0.99 and 0.93 when $P_{\max}$ equals 14 and 12, respectively, while maximizing throughput (which was calculated through exhaustive search).

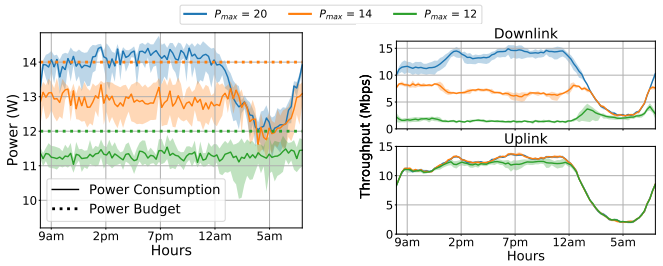
6.4 Comparison with other approaches

We complete our evaluation comparing our solutions with a state-of-the-art deep reinforcement learning algorithm: the Deep Deterministic Policy Gradient (DDPG) [68]. This algorithm needs to be customized since it is designed to solve the full-RL problem while in this work we face a contextual bandit problem. There are two main differences between these two problems. First, the full-RL considers



(a) (b)

Fig. 12: Performance evaluation of BP-vRAN throughout one day.



(a) (b)

Fig. 13: Performance evaluation of SBP-vRAN throughout one day.

that selected actions (control policies) have an impact on futures states (contexts). This assumption does not hold in our setting since the configuration of the vBS does not affect future contexts (traffic load and channel quality of the users). Second, in the full-RL problem, the reward can be delayed over time, while in our setting the performance is available at the end of the decision period.

The DDPG is implemented using an actor-critic deep neural network (NN) architecture and, in order to adapt it to the contextual bandit problem, we configure the critic NN to approximate the reward function instead of the Q-value function (see [17] for more details). We consider the same NN architecture as in [17] but we use a sigmoid as the activation function of the output layer of the actor NN. Since the action space of the DDPG is continuous (the output of the actor is a continuous vector with the same dimensions as $\mathcal{X}$), the selected actions are cast to the closer control policies that can be configured by the vBS. Moreover, we optimize the hyperparameters to minimize convergence time. Our experiments show that the DDPG converges to the same solutions as the proposed Bayesian-based algorithms, but lacks in convergence speed and versatility. We illustrate these issues using both problems that we presented in Sec. 4.2 and 4.3 and one context, as in Sec. 6.2.

For the first problem (Sec. 4.2), we configure the reward function of the DDPG to be the objective function in eq. (3). Fig. 14 shows the time evolution of the objective function for BP-vRAN and DDPG, for different values of b. Notably, DDPG converges to the same optimal policy learned by BP-vRAN but *has to invest one order of magnitude longer time*. The main reason for this difference is that our approach infers correlations in the objective function over the context-action

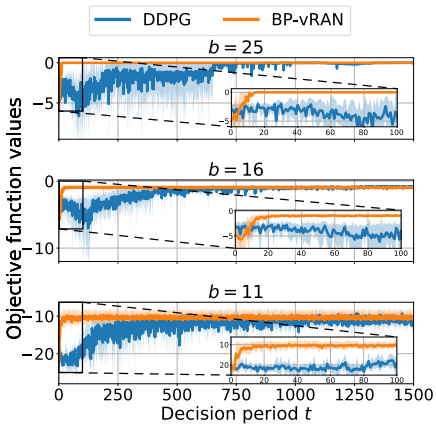


Fig. 14: Comparison of BP-vRAN with the customized DDPG

space more efficiently; and hence finds optimal policies even for unseen context-action pairs. This highlights the *data-efficiency* of the GP-based solution. It is also worth reminding that, differently to our benchmark, BP-vRAN has mathematical guarantees in performance (see Sections 5.1).

In order to implement the constrained problem in Sec. 4.3, we consider a customized reward function for the DDPG. The reward is encoded using a step function that takes the value of eq. (2) when the observed power is below P_{max}, and the minimum reward value otherwise. Fig. 15 shows the evolution over time of the power consumption and the associated throughput performance of the vBS for SBP-vRAN and DDPG. We begin the experiment by setting the power constraint equal to 15W, and changing it to 13W at decision period $t = 2000$.

Our results render three observations: (i) SBP-vRAN attains considerable convergence improvements over its benchmark (roughly, an order of magnitude). (ii) SBP-vRAN is unaffected by a sudden change on the power constraint; note that it only requires the change of P_{max} in line 5, Algorithm 2. Conversely, DDPG needs to change the configuration of the step function, which forces to restart its learning process from scratch, failing the hard constraint until decision period 3500, approximately. (iii) DDPG cannot perform safe exploration: it must *use* policies that violate the power constraint to *learn* so. On the other hand, our approach computes the uncertainty of each estimation, which allows us to implement safe exploration and satisfy the constraint with high probability. (iv) Although the DDPG can potentially find better solutions due to its continuous action space, our results show that both approaches converge to the same solution due to the fine-grained discretization of the action space of BP-vRAN and SBP-vRAN. Finally, it is important to remark the inherent drawback of GP-based approaches is the involved $\mathcal{O}(N^3)$ computation complexity (for Cholesky decomposition) in each orchestration period, where N is the number of data points. We observed in our experiments, however, that the unprecedented convergence speed of these methods pays off in a very short time. Moreover, we found that these computations do not induce a delay since, according to O-RAN specifications, there is a wide-enough time window to update the policy.

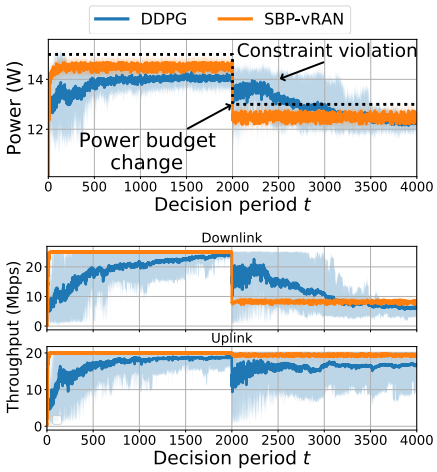


Fig. 15: Comparison of SBP-vRAN with the customized DDPG

7 Conclusions

The goal of this paper was threefold. First, to conduct an in-depth experimental study of the power consumption of virtualized base stations (vBSs); secondly, to propose two Bayesian learning algorithms that optimize the vBS performance subject to power constraints; and thirdly to evaluate these algorithms in realistic conditions using a fully-fledged wireless testbed, and compare them with state-of-the-art solutions that use deep neural networks.

Our findings revealed an intricate relationship between performance, power consumption, and key vBS control knobs, which renders impractical traditional resource control policies and motivate machine-learning solutions. Moreover, we saw that Bayesian learning algorithms can indeed enable efficient vBS operation; yet they require extensions and amendments in order to account for the network context and other practical and problem-specific issues. Finally, we found that these approaches are more data-efficient than state-of-the-art deep reinforcement learning solutions, but are also more computationally-demanding. This latter property does not pose a problem for O-RAN systems, according to their operation requirements, but might become a limitation for other resource control problems running in finer time granularity – yet, there are remedies that can reduce the computing load, e.g., re-initializing the GP approximation.

The considered problems are motivated by the latest industry developments in next generation virtualized RANs, and are centered around power consumption which is probably their most prevalent design constraint. Similarly, our solutions are in line with the requirements for automated, data-driven, platform-oblivious vRAN configuration. As such, we believe this work opens a new research direction and to that end we also make publicly available our testbed implementations and the collected measurements. We have released the source code of BP-vRAN and SBP-vRAN along with the dataset used in this work to foster future research in this area.

Acknowledgements

This work has been supported by the European Commission through Grant No. 101017109 (DAEMON project) and Grant No. 856709 (5Growth); and by the CERCA Programme/Generalitat de Catalunya.

References

[1] "AT&T and Nokia Accelerate the Deployment of RAN Open Source," AT&T. Press Release, 2019. [Online]. Available: https://about.att.com/story/2019/open_source.html
[2] I. Gomez-Miguelez *et al.*, "srsLTE: an Open-source Platform for LTE Evolution and Experimentation," in *in Proc. of ACM WinTech*, 2016.
[3] N. Nikaein *et al.*, "OpenAirInterface: A flexible platform for 5G research," *ACM SIGCOMM CCR*, vol. 44, no. 5, pp. 33–38, 2014.
[4] "Virtualized Radio Access Network: Architecture, Key Technologies and Benefits," Samsung. Technical Report, 2019.
[5] "Open & Virtualized – The Future of Radio Access Network," NEC. White Paper, 2020.
[6] O-RAN Alliance, "O-RAN-WG1-O-RAN Architecture Description - v01.00.00." Technical Specification, Februry 2020.
[7] "Reimagining the End-To-End Mobile Network in the 5G Era," Cisco, Rakuten, Altiostar. White Paper, 2019.
[8] "5G network energy efficiency," Nokia Corporation. White Paper, 2016.
[9] T. Hatt and E. Kolta, "5g energy efficiencies: green is the new black," GSMA Intelligence, Tech. Rep., 2020.
[10] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. d. Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.
[11] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT Press, 2006, vol. 2, no. 3.
[12] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian Processes," in *Proc. of ICML*, 2015, pp. 997–1005.
[13] Y. Sui *et al.*, "Stagewise Safe Bayesian Optimization with Gaussian Processes," *arXiv preprint arXiv:1806.07555*, 2018.
[14] D. Bega *et al.*, "CARES: Computation-aware Scheduling in Virtualized Radio Access Networks," *IEEE Trans. on Wireless Communications*, vol. 17, no. 12, pp. 7993–8006, 2018.
[15] D. Raca *et al.*, "On Leveraging Machine and Deep Learning for Throughput Prediction in Cellular Networks: Design, Performance, and Challenges," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 11–17, 2020.
[16] N. Zhao *et al.*, "Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Cellular Networks," *IEEE Trans. on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, 2019.
[17] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, and J. J. Alcaraz, "vrAIn: Deep Learning based Orchestration for Computing and Radio Resources in vRANs," *IEEE Transactions on Mobile Computing*, 2020.
[18] T. P. Lillicrap, et al., "Continuous Control with Deep Reinforcement Learning," in *Proc. of ICLR*, 2016.
[19] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Bayesian online learning for energy-aware resource orchestration in virtualized rans," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021.
[20] P. Rost *et al.*, "Computationally Aware Sum-Rate Optimal Scheduling for Centralized Radio Access Networks," in *Proc. of IEEE GLOBECOM*, 2015.
[21] K. Wang *et al.*, "Computing Aware Scheduling in Mobile Edge Computing System," *Springer Wireless Networks*, pp. 1–17, 2019.
[22] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2224–2287, 2019.
[23] D. Bega *et al.*, "DeepCog: Optimizing Resource Provisioning in Network Slicing With AI-Based Capacity Forecasting," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 361–376, 2020.
[24] H. Guo, J. Liu, and J. Lv, "Toward intelligent task offloading at the edge," *IEEE Network*, vol. 32, no. 2, pp. 128–134, 2020.

[25] A. Galanopoulos, J. A. Ayala-Romero, G. Iosifidis, and D. Leith, "Bayesian online learning for mec object recognition systems," in *2020 IEEE Global Communications Conference (Globecom)*. IEEE, 2020.

[26] J. A. Ayala-Romero, J. J. Alcaraz, A. Zanella, and M. Zorzi, "Online learning for energy saving and interference coordination in hetnets," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1374–1388, 2019.

[27] N. Liakopoulos, G. Paschos, P. Mertikopoulos, "No Regret in Cloud Resources Reservation with Violation Guarantees," in *Proc. of IEEE INFOCOM*, 2019.

[28] V. Valls, G. Iosifidis, G. de Mel, and L. Tassiulas, "Online network flow optimization for multi-grade service chains," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1329–1338.

[29] F. Mismar, J. Choi, and B. L. Evans, "A Framework for Automated Cellular Network Tuning With Reinforcement Learning," *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7152–7167, 2019.

[30] J. J. Alcaraz, J. A. Ayala-Romero, J. Vales-Alonso, and F. Losilla-López, "Online reinforcement learning for adaptive interference coordination," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 10, p. e4087, 2020.

[31] Z. Zhang, L. Ma, K. Poularakis, K. K. Leung, and W. Lingfei, "Dq scheduler: Deep reinforcement learning based controller synchronization in distributed sdn," in *Proceedings of IEEE ICC*, 2019.

[32] I. Alqerm and B. Shihada, "Sophisticated online learning scheme for green resource allocation in 5g heterogeneous cloud radio access networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, pp. 2423–2437, 2018.

[33] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361–373, 2017.

[34] A. Bastian *et al.*, "CBA: Contextual Quality Adaptation for Adaptive Bitrate Video Streaming," in *Proc. of IEEE INFOCOM*, 2019.

[35] X. Wang, X. Guo, J. Chuai, Z. Chen, and X. Liu, "Kernel-based Multi-Task Contextual Bandits in Cellular Network Configuration," in *Proc. of IEEE Conf. on Big Data*, 2019.

[36] J. Chuai, Z. Chen, G. Liu, X. Guo, X. Wang, X. Liu, C. Zhu, and F. Shen, "A collaborative learning based approach for parameter configuration of cellular networks," in *Proceedings of IEEE INFOCOM*, 2019.

[37] M. Anjum Qureshi and C. Tekin, "Fast learning for dynamic resource allocation in ai-enabled radio networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 95–110, 2020.

[38] M. Hashemi, A. Sabharwal, C. E. Koksal, and N. B. Shroff, "Efficient Beam Alignment in Millimeter Wave Systems Using Contextual Bandits," in *Proc. of IEEE INFOCOM*, 2018.

[39] A. Krause and C. S. Ong, "Contextual Gaussian Process Bandit Optimization," in *Proc. of NIPS*, 2011, pp. 2447–2455.

[40] O. Alipourfard et al., "CherryPick: Adaptively Unearthing the Best Cloud Configurations for Big Data Analytics," in *Proc. of USENIX NSDI*, 2017.

[41] S. Chinchali, et al., "Cellular Network Traffic Scheduling with Deep Reinforcement Learning," in *Proc. of AAAI*, 2018.

[42] Y. S. Nasir, and D. Guo, "Deep Actor-Critic Learning for Distributed Power Control in Wireless Mobile Networks," in *Proc. of Asilomar*, 2020.

[43] S. Bhaumik *et al.*, "CloudIQ: A Framework for Processing Base stations in a Data Center," in *Proc. of ACM Mobicom*, 2012.

[44] W. Wu *et al.*, "PRAN: Programmable Radio Access Networks," in *Proc. of ACM HotNets*, 2014.

[45] P. Rost, S. Talarico, and M. C. Valenti, "The complexity–rate tradeoff of centralized radio access networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6164–6176, 2015.

[46] T. X. Tran *et al.*, "Understanding the Computational Requirements of Virtualized Baseband Units Using a Programmable Cloud Radio Access Network Testbed," in *Proc. of IEEE ICAC*, 2017.

[47] N. Nikaein, "Processing Radio Access Network Functions in the Cloud: Critical Issues and Modeling," in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, 2015, pp. 36–43.

[48] G. Auer *et al.*, "How Much Energy is Needed to Run a Wireless Network?" *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40–49, 2011.

[49] H. Holtkamp *et al.*, "A parameterized base station power model," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2033–2035, 2013.

[50] O. Arnold, F. Richter, G. Fettweis, and O. Blume, "Power Consumption Modeling of Different Base Station Types in Heterogeneous Cellular Networks," in *IEEE Future Netw. & Mob. Summit*, 2010.

[51] M. Deruyck, W. Joseph, and L. Martens, "Power Consumption Model for Macrocell and Microcell Base Stations," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 3, pp. 320–333, 2014.

[52] B. H. Jung, H. Leem, and D. K. Sung, "Modeling of Power Consumption for Macro-, Micro-, and RRH-based Base Station Architectures," in *Proc. of IEEE VTC*, 2014, pp. 1–5.

[53] C. Desset *et al.*, "Flexible power modeling of lte base stations," in *Proc. of IEEE WCNC*, 2012.

[54] B. Debaillie *et al.*, "A Flexible and Future-proof Power Model for Cellular Base Stations," in *Proc. of IEE VTC*, 2015.

[55] N. Budhdev, M. C. Chan, and T. Mitra, "Pr 3: Power efficient and low latency baseband processing for lte femtocells," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2357–2365.

[56] T. Zhao, J. Wu, S. Zhou, and Z. Niu, "Energy-delay tradeoffs of virtual base stations with a computational-resource-aware energy consumption model," in *Proc. of IEEE ICCS*, 2014.

[57] J. Mo, and J. Walrand, "Fair End-to-End Window-Based Congestion Control," *IEEE/ACM Trans. on Netw.*, vol. 8, no. 5, pp. 556–567, 2000.

[58] L. Diez, A. Garcia-Saavedra, V. Valls, X. Li, X. Costa-Perez, and R. Aguero, "LaSR: A supple multi-connectivity scheduler for multi-RAT OFDMA systems," *IEEE Transactions on Mobile Computing*, 2018.

[59] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A Contextual-bandit Approach to Personalized News Article Recommendation," in *Proc. of ACM WWW*, 2010, pp. 661–670.

[60] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *Math. of Oper. Research*, vol. 35, no. 2, pp. 395–411, 2010.

[61] D. Duvenaud, "Automatic model construction with gaussian processes," Ph.D. dissertation, University of Cambridge, 2014.

[62] A. D. Bull, "Convergence Rates of Efficient Global Optimization Algorithms," *Journal of Machine Learn. Res.*, vol. 12, pp. 2879–2904, 2011.

[63] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design," in *Proceedings of ICML*, 2010.

[64] S. Amani *et al.*, "Regret Bounds for Safe Gaussian Process Bandit Optimization," *arXiv preprint arXiv:2005.01936*, 2020.

[65] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics," *arXiv preprint arXiv:1602.04450*, 2016.

[66] M. Fiducioso, S. Curi, B. Schumacher, M. Gwerder, and A. Krause, "Safe Contextual Bayesian Optimization for Sustainable Room Temperature PID Control Tuning," *arXiv preprint arXiv:1906.12086*, 2019.

[67] C. Márquez, M. Gramaglia, M. Fiore, A. Banchs, and Z. Smoreda, "Identifying Common Periodicities in Mobile Service Demands with Spectral Analysis," in *Proc. of MedComNet*, 2020.

[68] T. P. Lillicrap, et. al, "Continuous Control with Deep Reinforcement Learning," in *Proc. of ICLR*, 2016.

APPENDIX

This section provides details for the proof of Lemma 1, that is based on [63] and [39].

Lemma 2. *Let $\epsilon \in (0,1)$, assume that the noise in the observation is uniformly bounded by ζ and $\beta_t = 2B^2 + 300\gamma_t \ln^3(t/\epsilon)$, then:*

$$Pr\{\forall t,\ \forall z \in \mathcal{Z}, |\mu_{t-1}(z) - u(z)| \leq \sqrt{\beta_t}\sigma_{t-1}(z)\} \geq 1 - \epsilon. \tag{16}$$

Proof. Given in [63, Theorem 6]. □

Lemma 3. *Fix $t \geq 1$. If $|u(z) - \mu_{t-1}(z)| \leq \sqrt{\beta_t}\sigma_{t-1}(z)$ for all $z \in \mathcal{Z}$, the contextual regret r_t is then bounded by $2\sqrt{\beta_t}\sigma_{t-1}(z_t)$.*

Proof. The proof follows [39, Lemma 4.1]. Let $x_t^* \in \text{argsup}_{x\in\mathcal{X}} u(\omega_t, x)$ be the optimal control at decision period t. Then, considering x_t to be the selected control at decision period t, given the acquisition function in eq 11: $\mu_{t-1}(\omega_t, x_t) + \sqrt{\beta_t}\sigma_{t-1}(\omega_t, x_t) \geq \mu_{t-1}(\omega_t, x_t^*) + \sqrt{\beta_t}\sigma_{t-1}(\omega_t, x_t^*) \geq u(\omega_t, x_t^*)$. Therefore, $r_t = u(\omega_t, x_t^*) - u(\omega_t, x_t) \leq \sqrt{\beta_t}\sigma_{t-1}(\omega_t, x_t) + \mu_{t-1}(\omega_t, x_t) - u(\omega_t, x_t) \leq 2\sqrt{\beta_t}\sigma_{t-1}(\omega_t, x_t)$. □

Lemma 4. *The information gain for the points selected can be expressed in terms of the predictive variances. If $u_T = (u(z_t)) \in \mathbb{R}^T$ [63]:*

$$I(y_T; u_T) = \frac{1}{2}\sum_{t=1}^{T} \log(1 + \zeta^{-2}\sigma_{t-1}^2(z_t)) \tag{17}$$

Proof. Given in [63, Lemma 5.3]. □

Proof of Lemma 1. The proof follows [39, Theorem 5]. By Lemma 2 and Lemma 3 we have that $Pr\{r_t^2 \leq 4\beta_t\sigma_{t-1}^2(z_t)\forall t \geq 1\} \geq 1-\epsilon$. Given that β_t is non-decreasing, we have that

$$\begin{aligned} 4\beta_t\sigma_{t-1}^2(z_t) &\leq 4\beta_T\zeta^2(\zeta^{-2}\sigma_{t-1}^2(z_t)) \\ &\leq 4\beta_T\zeta^2 C_2 \log(1 + \zeta^{-2}\sigma_{t-1}^2(z_t)) \end{aligned} \tag{18}$$

with $C_2 = \zeta^{-2}/\log(1+\zeta^{-2}) \geq 1$, since $s^2 \geq C_2\log(1+s^2)$ for $s \in [0, \zeta^{-2}]$, and $\zeta^{-2}\sigma_{t-1}^2(z_t) \leq \zeta^{-2}k(z_t, z_t) \leq \zeta^{-2}$. Considering $C_1 = 8\sigma^2 C_2$ and $R_T^2 \leq T\sum_{t=1}^{T} r_t^2$ (Cauchy-Schwarz inequality), the result follows from Lemma 4. □