Sample-efficient Learning for Edge Resource Allocation and Pricing with BNN Approximators

Feridun Tütüncüoğlu and György Dán
Division of Network and Systems Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
Email: {feridun, gyuri}@kth.se

***Abstract*—Edge computing (EC) is expected to provide low latency access to computing and storage resources to autonomous Wireless Devices (WDs). Pricing and resource allocation in EC thus have to cope with stochastic workloads, on the one hand offering resources at a price that is attractive to WDs, one the other hand ensuring revenue to the edge operator. In this paper, we formulate the strategic interaction between an edge operator and WDs as a Bayesian Stackelberg Markov game. We characterize the optimal strategy of the WDs that minimizes their costs. We then show that the operator's problem can be formulated as a *Markov Decision Process* and propose a model-based reinforcement learning approach, based on a novel approximation of the workload dynamics at the edge cell environment. The proposed approximation leverages two *Bayesian Neural Networks* (BNNs) to facilitate efficient policy learning, and enables sample efficient transfer learning from simulated environments to a real edge environment. Our extensive simulation results demonstrate the superiority of our approach in terms of sample efficiency, outperforming state-of-the-art methods 30 times in terms of learning rate and by 50% in terms of operator revenue.**

***Index Terms*—Edge computing, Markov decision process, Bayesian neural networks**

I. Introduction

Edge Computing (EC) brings computational resources close to the network edge, facilitating on-demand offloading of latency sensitive and computationally intensive tasks from *Wireless Device*s (WDs). The commercial deployment of edge computing faces technological and financial challenges, however. First, memory, computing and storage resources are constrained compared to cloud offerings, which is problematic considering the ambition to serve a varying amount of latency sensitive tasks. Second, EC requires a pricing model that can adapt to varying demand, is transparent to the WDs, and is easy to deploy. Third, the parameters' of the WDs and their workloads may not be shared with the edge operator due to privacy concerns and to keep communication overhead low [1].

Addressing these challenges requires efficient resource management algorithms that enable the deployment of commercially viable adaptive pricing schemes under information asymmetry. A common approach in the literature to address the above problem is to formulate an optimization problem and to develop algorithmic solutions. This approach becomes either computationally infeasible or it has to be based on assumption that are difficult to uphold in practice [1]–[3]. Another approach is to follow a game theoretic treatment, e.g., formulating the problem as an auction [4] or as a Stackelberg game [5], but these approaches do not scale well either.

The alternative approach is to formulate the problem as sequential decision making problem under uncertainity and use model-free reinforcement learning, where a policy is learnt through interactions with the environment, which typically requires many interactions for convergence, i.e., it has low sample efficiency [6], [7]. Hence, how to provide an efficient solution for pricing and resource allocation under stochastic workloads and information asymmetry is an open problem.

In this work, we address this problem by proposing a model-based learning approach. Our approach is based on training *Bayesian Neural Network* (BNN) approximators of the environment, and then training a resource allocation and pricing policy through interaction with the BNN approximators, without the need for sampling from the real edge environment to explore new policies. As our results show, this approach has high sample efficiency and at the same time it ensures high average revenue over time periods of practical interest.

The rest of the paper is organized as follows. We describe the system model and the problem formulation in Section II. We characterize the WDs's best response in Section III. In Section IV, we propose an MDP formulation, and we present our proposed model-based *Reinforcement Learning* (RL) solution in Section V. We provide numerical results in Section VI. We discuss the related work in Section VII. We conclude the paper in Section VIII.

II. System Model

We consider an EC system that consists of an edge server with storage capacity S (in Bytes), compute capacity F (in *Instructions per Second* (IPS)) and communication capacity W (in Hz). The EC system provides *Function as a Service* (FaaS) (also known as serverless computing) to a dynamic population of WDs. There is a set $\mathcal{J}$ of applications, and the edge operator can cache a subset $\mathcal{X} \subseteq \mathcal{J}$ of the applications, e.g., the software images required for the execution of the tasks. The set $\mathcal{X}$ of cached applications has to satisfy the storage capacity constraint $\sum_{j \in \mathcal{X}} s_j \leq S$ where s_j is the size of application image j.

The work was partly funded by the Vinnova Center for Trustworthy Edge Computing Systems and Applications (TECoSA) and the Swedish Research Council through project 2020-03860.

A. User Model

We denote by T_i the arrival time of WD i to the EC system, and by $\phi_i \in \mathcal{J}$ the type of WD $i \in \mathbb{N}^+$, i.e., the application it would like to execute. The computational task of WD i is characterized by the amount of input data d_i, by the average number of *instructions* (I) per byte L_{ϕ_i} required to perform the task, and by the completion time requirement $\overline{\tau}_i^l$. We assume that $d_i \sim D_{\phi_i}$ is an i.i.d. random variable, known to WD i. We consider that the arrivals of WDs of type $j \in \mathcal{J}$ can be modeled by a homogeneous process $\mathcal{B}_j$ with rate Λ_j. The arrival processes for different types of WDs are independent.

If application ϕ_i, which WD i intends to use, is cached by the operator ($\phi_i \in \mathcal{X}$) then WD i can decide whether to offload the computation to the edge server. We denote by o_i the offloading decision of WD i: $o_i = 1$ corresponds to offloading and $o_i = 0$ to local computing. If WD i chooses not to offload or its application is not cached by the operator, then it departs the edge system and performs its task locally. In what follows we define the cost model of WD i with and without offloading.

1) Local Computing: If WD i chooses not to offload, the task needs to be executed using local computing resources (i.e., local CPU). We denote by f_i^l the local processing power (in IPS) of WD i and we use it to express the local processing time

$$\tau_i^l = \frac{L_{\phi_i} d_i}{f_i^l}. \tag{1}$$

We consider that f_i^l is chosen such that local computing completes upon the task completion deadline $\overline{\tau}_i^l$ of the task of WD i, i.e., $\tau_i^l = \overline{\tau}_i^l$. Thus, the task completion deadline $\overline{\tau}_i^l$ will influence the decision of the WD whether or not to offload. This assumption is reasonable, as dynamic voltage and frequency scaling are widely used for reducing the energy consumption of battery powered WDs while meeting performance needs [8].

2) Computation Offloading: If WD i decides to offload, it has to transmit d_i amount of data wirelessly to the edge server via an *Access Point* (AP), and then processing is performed at the edge server. We denote by w_i the *Bandwidth* (BW) allocated to WD i by the edge operator. Assuming a Gaussian channel [1], we can express the upload time of WD i by,

$$\tau_i^u(p_i, w_i) = \frac{d_i}{w_i \log_2(1 + \frac{p_i h_i}{\sigma_i^2})}, \tag{2}$$

where h_i is the channel coefficient from WD i to the AP, p_i is the transmit power of the WD i, and σ_i^2 is the noise power at the AP. We consider that the transmit power is bounded by the maximum transmit power $\overline{p}_i$, i.e. $p_i \leq \overline{p}_i$. This model of the transmission rate corresponds to *Orthogonal Frequency-division Multiple Access* (OFDMA), adopted in 5G and WiFi6, which avoids intra-cell interference by using non-overlapping subcarriers for data transmission [9]. Similar to previous works [10], we make the common assumption that the time needed to transmit the results of the computation from the edge server to the WD is negligible, because for many applications (e.g., object detection, recognition and etc.) the size of the output is significantly smaller than the size of the input data.

We denote by f_i the computing power of the edge server (measured in IPS) allocated to the task of WD i and we express the processing time at the edge server as

$$\tau_i(f_i) = \frac{L_{\phi_i} d_i}{f_i}. \tag{3}$$

If WD i decides to offload then the allocated compute and communication capacity (f_i and w_i) will be reserved for the WD from $[T_i, T_i + \tau_i^u(p_i, w_i) + \tau_i(f_i)]$, i.e., until WD i is done with offloading. Hence, the total bandwidth and compute allocation at any point in time cannot exceed W and F, respectively.

3) WD Cost Model: Computation and offloading incur energy consumption and monetary cost to the WDs. In case of local computing, the cost is the energy consumed by the WD for executing the task,

$$C_i^0 = \tau_i^l (f_i^l)^2 \kappa_i \gamma_i, \tag{4}$$

where κ_i is the energy efficiency parameter of WD i (measured in J per Hz per I^2) and γ_i is the unit local energy cost (measured in \$ per J). γ_i is determined by the cost of electricity and by the cost of charging the battery of WD i, e.g., in terms of time, etc. and serves as the conversion factor from energy consumption to monetary cost. We make the reasonable assumption that γ_i is known to WD i and thus C_i^0 can be computed.

In case of offloading, we define the offloading cost as the sum of the energy consumption cost for transmitting the input data and the price that is to be paid, i.e.,

$$C_i^1(p_i, f_i, w_i) = \tau_i^u(p_i, w_i) p_i \beta_i \gamma_i + \pi_{\phi_i} L_{\phi_i} d_i, \tag{5}$$

where β_i denotes the transmit antenna power efficiency parameter of WD i. Then the cost of WD i is

$$C_i(p_i, f_i, w_i, o_i) = (1 - o_i) C_i^0 + o_i C_i^1(p_i, f_i, w_i). \tag{6}$$

We consider that the WDs have a preference for saving the state of charge of their batteries, thus in case of a tie between local computing cost and offloading cost the WD would choose to offload. The local cost associated with a WD reflects its valuation of task execution, and its formulation aligns with the modeling approach previously employed in cloud computing [11]. In economic terms, this valuation is akin to the concept of the *reservation price*, which represents the maximum price a customer would be willing to pay for a specific product or service [12].

B. Problem Formulation

We consider that the WDs and the operator are rational, strategic entities. The objective of WD i is to minimize its cost subject to its completion time requirement, the constraint

on the maximum transmission power, and the caching decision of the operator. Thus, WD i aims to solve

$$\min_{p_i,\, o_i} \quad C_i(p_i, f_i, w_i, o_i) \tag{7}$$

$$s.t. \quad o_i(\tau_i^u(p_i, w_i) + \tau_i(f_i)) \leq \tau_i^l, \tag{8}$$

$$\mathbb{1}_{\phi_i \in \mathcal{X}} - o_i \geq 0, \tag{9}$$

$$p_i \leq \overline{p}_i, \tag{10}$$

where the first constraint ensures that WD i does not offload if $\tau_i^u(p_i, w_i) + \tau_i(f_i) > \tau_i^l$, i.e., if the completion time when offloading exceeds the completion deadline, the second constraint ensures that WD i offloads only if application ϕ_i is cached by the operator and $\mathbb{1}_{(.)}$ is the indicator function, and the last constraint ensures that the transmit power does not exceed the maximum transmit power.

Aligned with FaaS pricing models used today, we consider that the income of the operator depends on the price it sets for offloading and on whether or not WDs decide to offload. We model the interaction as follows. Upon arrival of WD i, the operator offers compute resources f_i, communication resources w_i and offloading price π_{ϕ_i} to the WD. Thus the income that the operator gets from WD i is

$$U_{i,\mathcal{X}} = o_i L_{\phi_i} d_i \pi_{\phi_i}. \tag{11}$$

We consider that the operator allocates resources subject to utilization upon arrival i and the capacity, i.e., $F_i^\Sigma + f_i \leq F$ and $W_i^\Sigma + w_i \leq W$ where we denote by $F_i^\Sigma = \sum_{i \in \mathcal{N}_i^o} f_i$ the sum of compute power allocations at time T_i, and by $W_i^\Sigma = \sum_{i \in \mathcal{N}_i^o} w_i$ the sum of BW allocations at time T_i, and let $\mathcal{N}_i^o$ be the set of offloaders at time T_i.

We consider that the operator aims at maximizing its utility by choosing a resource allocation and pricing policy $\theta_\mathcal{X}$ for cached applications $\mathcal{X}$, i.e., the operator wants to solve

$$\theta_\mathcal{X}^* = \underset{\theta_\mathcal{X} \in \Theta_\mathcal{X}}{\operatorname{argmax}} \mathbb{E}_{\theta_\mathcal{X}} \Big[\sum_{i=0}^{\infty} \zeta^i U_{i,\mathcal{X}} \Big], \tag{12}$$

where $\zeta = e^{-\beta/\Lambda}$ is the discount factor for $\Lambda = \sum_{j \in \mathcal{J}} \Lambda_j$ and some $\beta > 0$. This form of discounting is a reasonable approximation of discounting in continuous time, and is accurate for $\beta << 1$ and $\Lambda >> 0$ since $\beta i / \Lambda \approx \beta T_i$.

We make the reasonable assumption that the WDs' parameters are private information, hence the operator cannot compute the offloading decision of WD i for given resource allocation and pricing (f_i, w_i, π_{ϕ_i}), but it has to learn the behavior of the WDs. The resulting problem can be modeled as a Bayesian Stackelberg Markov game with short-run players, i.e., a dynamic game where the operator (the leader) repeatedly engages in strategic interaction with one of $|\mathcal{J}|$ types of players, and the stage games affect subsequent stage games through the evolution of a state (the allocated resources). In what follows we analyze this game and propose a scalable solution through model-based RL.

III. User Best Response Characterization

We start the analysis with characterizing the best response of WD i in a stage game, i.e., for given caching decision $\mathcal{X}$, pricing π_{ϕ_i} and resource allocation f_i, w_i, announced by the operator. We first show that the best response has a threshold structure and can be computed efficiently by the WD.

Lemma 1. *Consider a WD i that arrives to the edge cell where its application is cached by the operator, i.e., $\phi_i \in \mathcal{X}$. If $\tau_i(f_i) > \tau_i^l$, then $o_i^* = 0$. Otherwise, let p_i^* be such that $\tau_i^u(p_i, w_i) + \tau_i(f_i) = \tau_i^l$. Then if $p_i^* > \overline{p}_i$, $o_i^* = 0$, otherwise,*

$$o_i^* = \begin{cases} 1, & \pi_{\phi_i} \leq \overline{\pi}_i, \\ 0, & else, \end{cases} \tag{13}$$

where $\overline{\pi}_i = f_i^l \kappa_i \gamma_i - p_i^* \beta_i \gamma_i (\frac{1}{f_i^l} - \frac{1}{f_i})$.

Proof. Observe that if $\tau_i(f_i) > \tau_i^l$, then WD i cannot complete the task on time if it offloads, thus to complete the task before the deadline it has to perform local computing, i.e., the optimal offloading decision is $o_i^* = 0$. Otherwise, WD i should choose a transmit power that minimizes its cost while ensuring timely completion. Observe that the uploading time $\tau_i^u(p_i, w_i)$ is a strictly monotonically decreasing function of p_i, and $C_i^1(p_i, f_i, w_i)$ is a strictly monotonically increasing function of p_i. Thus, WD i minimizes its cost by choosing a transmit power p_i^* that yields $\tau_i^u(p_i, w_i) + \tau_i(f_i) = \tau_i^l$. Now, if $p_i^* > \overline{p}_i$ then offloading is not feasible. Otherwise, if $p_i^* \leq \overline{p}_i$ then the optimal decision is to offload if and only if $C_i^1(p_i, f_i, w_i) \leq C_i^0$, i.e.,

$$o_i^* = \begin{cases} 1, & \tau_i^u(p_i^*, w_i) p_i^* \beta_i \gamma_i + \pi_{\phi_i} L_{\phi_i} d_i \leq \tau_i^l (f_i^l)^2 \kappa_i \gamma_i, \\ 0, & else. \end{cases} \tag{14}$$

Since, the optimal transmit power yields $\tau_i^u(p_i^*, w_i) + \tau_i(f_i) = \tau_i^l$, we can substitute $\tau_i^u(p_i^*, w_i) = \tau_i^l - \tau_i(f_i)$, (1) and (3) into (14), and obtain (13), which proves the result. □

Lemma 1 shows that for a given action (f_i, w_i, π_{ϕ_i}) of the operator, WD i can compute its optimal action o_i^* efficiently. The lemma further shows that the optimal transmit power p_i^* yields a completion time that is equal to the task completion deadline, which intuitively aligns with the trade-off between faster transmission rates and increased energy consumption associated with increased transmit power.

IV. *Markov Decision Process* (MDP) Formulation For a Fixed Caching Decision

Given the best response of the WDs, we now focus on the maximization of the operator's revenue. Observe that, given the WDs' best response and a belief about the distribution of WD types and parameters, the operator's problem can be modeled as a sequential decision making problem under uncertainty. We indeed first show that the operator's problem can be formulated as a MDP. A continuous state MDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, R, \mathcal{P}, \zeta, \mathcal{P}_0 \rangle$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the set of actions, $\mathcal{A}_s$ is the set of actions available in state s [13], $R_i(s_i, a_i)$ is the instantaneous reward at time

step i as a function of the state $s_i \in \mathcal{S}$ and the action $a_i \in \mathcal{A}_{s_i}$. $\mathcal{P}$ is the state transitions that describes the state transition probability densities $p[s_{i+1}|s_i, a_i], \forall s_i \in \mathcal{S}, \forall a_i \in \mathcal{A}$. Finally, $\mathcal{P}_0$ is the probability density function of the initial state.

Before formulating problem (12) as an MDP, we introduce some notation. Let $\mathcal{H}_i^o = (i', T_{i'}, \phi_{i'}, f_{i'}, w_{i'}, \pi_{\phi_{i'}})_{i' \in \mathcal{N}_i^o}$ be the collection of the parameters of the WDs that offload at time T_i. We denote by $T_i^d = T_i + \tau_i^u(p_i^*, w_i) + \tau_i(f_i)$ the departure time of WD i if it offloads. Furthermore, we let $\overline{N}^o$ be the maximum number of offloaders that the edge server can serve concurrently.

Next, we show that problem (12) can be formulated as an MDP, by considering the system state at the arrival epochs of the WDs, as follows.

Theorem 1. *Let $s_i = (\phi_i, d_i, F_i^\Sigma, W_i^\Sigma, T_i, \mathcal{H}_i^o)$, let $a_i = (f_i, w_i, \pi_{\phi_i})$, let $R_i(s_i, a_i) = O_i(\phi_i, d_i, a_i) L_{\phi_i} d_i \pi_{\phi_i}$ where $O_i(\phi_i, d_i, a_i) \in \{0, 1\}$ is a Bernoulli random variable. Then problem* (12) *is an MDP.*

Proof. Based on the state definition s_i, the state space $\mathcal{S}$: $|\mathcal{J}| \times \mathbb{R}^+ \times [0, F] \times [0, W] \times \mathbb{R}^{\overline{N}^o} \times [0, F]^{\overline{N}^o} \times [0, W]^{\overline{N}^o}$, and the action space is $\mathcal{A} : [0, F] \times [0, W] \times \mathbb{R}$. Observe from Lemma 1 that the optimal offloading decision is computed based on the given action a_i and the parameters of the WD i. However, the only WD parameters known by the operator upon the arrival are L_{ϕ_i} and d_i. Thus, the reward due to arrival i becomes $R_i(s_i, a_i) = O_i(\phi_i, d_i, a_i) L_{\phi_i} d_i \pi_{\phi_i}$. Next, we need to show that the probability of a state transition from s_i to state s_{i+1} when taking action a_i can be fully described by s_i and a_i, i.e., that the state transitions have the Markov property. Let $\mathcal{N}_i^d = \{i' \in \mathcal{N}_i^o | T_{i'}^d \leq T_{i+1}\}$ be the set of departed WDs between time $(T_i, T_{i+1}]$. As the operator takes action on every arrival, the amount of used computation and communication resources can be expressed as $F_{i+1}^\Sigma = F_i^\Sigma - \sum_{i' \in \mathcal{N}_i^d} f_{i'} + O_i(\phi_i, d_i, a_i) f_i$ and $W_{i+1}^\Sigma = W_i^\Sigma - \sum_{i' \in \mathcal{N}_i^d} w_{i'} + O_i(\phi_i, d_i, a_i) w_i$, respectively. The probability of transitioning from $(F_i^\Sigma, W_i^\Sigma, \mathcal{H}_i^o)$ to $(F_{i+1}^\Sigma, W_{i+1}^\Sigma, \mathcal{H}_{i+1}^o)$ when taking action a_i can be expressed as

$$\begin{aligned} &\mathbb{P}[F_{i+1}^\Sigma, W_{i+1}^\Sigma, \mathcal{H}_{i+1}^o | F_i^\Sigma, W_i^\Sigma, \mathcal{H}_i^o, a_i, T_{i+1}] = \\ &\mathbb{P}[O_i(\phi_i, d_i, a_i) = o_i | d_i, \phi_i, a_i, \mathcal{H}_i^o] \\ &\prod_{k \in \mathcal{N}_i^d} \mathbb{P}[T_k^d \leq T_{i+1} | T_k] \prod_{k \in \mathcal{N}_i^o \setminus \mathcal{N}_i^d} \mathbb{P}[T_k^d > T_{i+1} | T_k]. \end{aligned} \tag{15}$$

Thus, the transition probability from s_i to s_{i+1} when taking action a_i is

$$\begin{aligned} p[s_{i+1}|s_i, a_i] = \mathbb{P}[F_{i+1}^\Sigma, W_{i+1}^\Sigma, \mathcal{H}_{i+1}^o | F_i^\Sigma, W_i^\Sigma, \mathcal{H}_i^o, a_i, T_{i+1}] \\ \mathbb{P}[\phi_{i+1} | \mathcal{H}_i^o] p[d_{i+1} | \phi_{i+1}] p(T_{i+1}). \end{aligned} \tag{16}$$

Clearly, (16) shows that the state transitions have the Markov property, hence (12) is an MDP. □

Since (12) is a MDP, an optimal policy could be computed using a *model free* (MF) RL approach. Nonetheless, this approach is impractical due to the large state space of the resulting MDP. To overcome this issue, in what follows we propose an approach that learns an approximate model of the system dynamics for improving sample efficiency.

V. *Approximate Dynamic Resource Allocation and Pricing* (ADRAP)

Our proposed ADRAP algorithm is based on a novel model-based RL approach. Existing model-based RL approaches learn an approximate state transition probability function $\hat{p}[s_{i+1}|s_i, a_i]$ and an approximate reward function $\hat{R}_i(s_i, a_i)$, and use these for training a policy θ through simulated interactions, which is then deployed in the real environment [14]. Nonetheless, in the considered problem the state transition probabilities cannot be approximated easily due to the aleatoric uncertainty caused by the WDs' workloads.

A. Environment Model

We propose to approximate the distribution of five quantities to decribe the system dynamics: the interarrival time distribution, the type ϕ_i of the next WD, the amount of data D_i, the task completion time $\overline{\tau}_i^l$ as a function of the action f_i, w_i and the offloading decision $O_i(\phi_i, d_i, a_i)$ of a WD as a function of the allocation f_i, w_i and the offered π_{ϕ_i}.

Recall that the arrival process $\mathcal{B}_j$ is assumed to be homogeneous and the input size D_i is i.i.d, hence the corresponding samples can be generated directly using inverse transform sampling based on the empirical CDFs of the collected data. The completion time and the decision to offload depend, however, on the decision of the operator. To deal with the dependence of these random variables on f_i, w_i and the π_{ϕ_i}, we propose to use two BNNs as approximators. We approximate the completion time $\hat{\tau}_i = f_\tau(x_i^\tau, z_i; \mathcal{W}^\tau) + \epsilon_i^\tau$ where $f_\tau(x_i^\tau, z_i; \mathcal{W}^\tau)$ is the output of a BNN trained to estimate the task completion time $\hat{\tau}_i$ for given input $x_i^\tau = [\phi_i, d_i, f_i, w_i]$, and z_i is a random disturbance with prior $z_i \sim \mathcal{N}(0, \Gamma_z)$, lastly, $\epsilon_i^\tau \sim \mathcal{N}(0, \tilde{\sigma}^\tau)$ is Gaussian noise. Similarly, we approximate the offloading decision $\mathbb{P}[O_i(\phi_i, d_i, a_i) = 1] = \frac{1}{N_d} \sum_{k=1}^{N_d} f_o(x_i^o, z_k; \mathcal{W}^o) + \epsilon_k^o$, where $f_o(x_i^o, z_i; \mathcal{W}^o)$ is the output of a BNN trained to estimate the offloading probability for given input $x_i^o = (\phi_i, d_i, f_i, w_i, \pi_i)$, and $\epsilon_k^o \sim \mathcal{N}(0, \tilde{\sigma}^o)$ is Gaussian noise. For brevity, we omit the derivation of the posterior and of the energy function, and refer to [14]–[16] for details.

B. Model-based RL

Our proposed algorithm is presented in Algorithm 1. We first choose a low dimensional state representation $s_i = (\phi_i, d_i, F_i^\Sigma, W_i^\Sigma)$ for our approximate MDP model. Initially (Line 2) we employ a random policy to take N_{tra} transition samples from the environment, which constitute the training data $\mathcal{D}$. The training data $\mathcal{D}$ are subsequently used for training the BNN approximators (Line 3) using stochastic gradient descent. We then use the BNN approximators with a model-free RL algorithm (Line 4) for learning policy θ^*, which the operator employs in the real environment.

Algorithm 1: ADRAP Algorithm

Initialize the training buffer $\mathcal{D}$ and a random policy θ
$\mathcal{D} \leftarrow$ *ENV-INTERACT*(N_{tra})
$\mathcal{W}^r, \mathcal{W}^\tau \leftarrow$ *TRAIN-BNN*$(\mathcal{D})$
$\theta^* \leftarrow$*FICTIONAL-TRAIN*$(\mathcal{W}^o, \mathcal{W}^\tau, N_{tra})$

---

procedure FICTIONAL-TRAIN$(\mathcal{W}^o, \mathcal{W}^\tau, N_{tra})$
  for $i \leq N_{tra}$ **do**
    $T_i \sim \hat{\mathcal{B}}_{\phi_i}$ ▷ Generate arrival time from the estimated arrival process
    $D_i \sim \hat{D}_{\phi_i}$ ▷ Sample from the estimated input size distribution
    $s_i = (\phi_i, d_i, F_i^\Sigma, W_i^\Sigma)$ ▷ WD i arrives
    Take action $\theta(s_i) \rightarrow a_i = (f_i, w_i, \pi_{\phi_i})$
    $R_i(s_i, a_i) = 0$
    $P_o(\phi_i, d_i, a_i) = \frac{1}{N_d} \sum_{k=1}^{N_d} f_o(x_i^o, z_k; \mathcal{W}^o) + \epsilon_k^o$
    if $P_o(\phi_i, d_i, a_i) \geq \mathcal{U}(0,1)$ **then**
      $F_{i+1}^\Sigma = F_i^\Sigma + f_i, \quad W_{i+1}^\Sigma = W_i^\Sigma + w_i$
      $R_i(s_i, a_i) = L_{\phi_i} d_i \pi_{\phi_i}$
      $\hat{\tau}_i = f_\tau(x_i^\tau, z_i; \mathcal{W}^\tau) + \epsilon_i^\tau$
    end if
    `// Handle departures`
    Update θ based on (s_i, a_i) using a MF approach
  end for
end procedure

VI. Numerical Results

We used extensive simulations to evaluate the performance of the proposed ADRAP algorithm.

We consider an edge system with storage capacity $S = 256$ GB, and compute capacity $F = 10000$ GIPS, corresponding to small compute cluster. The bandwidth is $W = 100$ MHz, which is typical in sub-6 GHz 5G deployments. The operator caches 3 applications, and we consider four workload scenarios that differ in the arrival intensities of the types of WDs. We refer to these as Cells #1 to #4. The task complexities, data sizes and inter-arrival time distributions at the cells are shown in Table I.

The operator allocates $f_i \in [200, 1000]$ GIPS and $w_i \in [0.1, 20]$ MHz to a WD, while the price $\pi_{\phi_i} \in [10^{-6}, 10^{-3}]$ \$/I. The maximum transmission power $\overline{p}_i$ of the WDs is drawn from a uniform distribution on $[0.1, 1]$ W and f_i^l is uniform on $[1, 10]$ GIPS. The channel channel gain h_i and the noise variance σ_i^2 are uniformly distributed on $[0.3, 1]$ and $[0.1, 1]$, respectively. The energy efficiency parameter κ_i and the unit energy cost parameter β_i are 10^{-2} J/Hz/I^2, and on 10^{-3}, respectively. We set $\gamma_i = 10^{-2}$ \$/J. These choices of parameters are similar to those used in [17].

The hyperparameters and output activation functions of the BNNs are shown in Table II. In order to train the BNNs we have collected 10000 transition samples for Cells $\#2, \#3, \#4$ and 30000 transition samples for Cell $\#1$ as the arrival intensity of App 0 in Cell $\#1$ is relatively low. We used the ADAM optimizer [18] to find with learning rates 0.005 and 0.03 and 600 training epochs for BNNs $f_o(.,.;\mathcal{W}^o)$ and $f_\tau(.,.;\mathcal{W}^\tau)$ respectively.

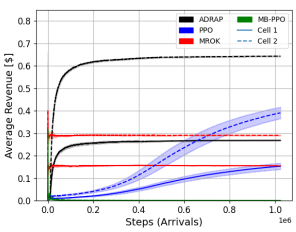
Cell	j	L_j [KI/GB]	D_j [GB]	Type [s]
#1	App 1	400	$\mathcal{U}(0.01, 0.1)$	Exp(2000)
	App 2	100	$\mathcal{U}(0.2, 0.5)$	Exp(1000)
	App 3	3	$\mathcal{U}(0.01, 0.1)$	Exp(20)
#2	App 1	400	$\mathcal{U}(0.01, 0.1)$	Exp(200)
	App 2	100	$\mathcal{U}(0.2, 0.5)$	Exp(1000)
	App 3	3	$\mathcal{U}(0.01, 0.1)$	Exp(20)
#3	App 1	40	$\mathcal{U}(0.01, 0.1)$	Exp(200)
	App 2	10	$\mathcal{U}(0.2, 0.5)$	Exp(1000)
	App 3	3	$\mathcal{U}(0.01, 0.1)$	Exp(20)
#4	App 1	40	$\mathcal{U}(0.01, 0.1)$	Γ(10,20)
	App 2	10	$\mathcal{U}(0.2, 0.5)$	Γ(100,10)
	App 3	3	$\mathcal{U}(0.01, 0.1)$	Γ(20,1)

TABLE I: Workload parameters in the considered scenarios.

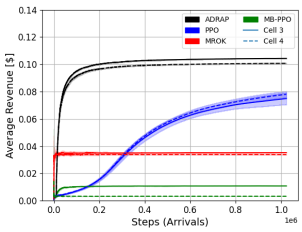
We consider three baselines for the evaluation. The first baseline is *Proximal Policy Optimization* (PPO), a model-free RL algorithm [19], with the state definition employed in ADRAP. We do not provide results for PPO with state definition given in Theorem 1 as it converges in the order of years, which is clearly not feasible. The second baseline is the pricing scheme for the *Multi-Resource Online Knapsack* (MROK) problem proposed in [11], which is a user-specific pricing scheme based on the instantaneous system load. The pricing scheme assumes one type of allocation of the resources, as these showed the best performance. The third baseline is a state of the art model based RL approach (*Model Based*-PPO (MB-PPO)), where two BNNs are used to estimate the next state and the reward for training the PPO policy. The results shown are the averages of at least 10 simulations per cell environment, together with 95% confidence intervals.

A. Operator Revenue

Fig. 1a and Fig. 1b show the operator's average revenue as a function of the number of arrivals (i.e., time), corresponding to a time intervals of approximately 8 months. The results show that ADRAP can well approximate the real environment model already after ten thousand arrivals, and hence the average revenue of ADRAP is about 50% higher than that of the baselines and provide up to 30 times faster learning compared to PPO. Among the baselines, MROK outperforms PPO and MB-PPO in terms of learning rate and performance. This is because MROK only chooses a price, hence its complexity is lower than that of the learning-based solutions. Comparing the results for different cells, we can observe that the revenue obtained in Cell #2 is significantly higher than that in Cell #1 as the arrival rate of high complexity applications (App 1 and 2) is higher, allowing the operator to charge more. Similarly, Cell #2 has higher revenue than Cell #3 and Cell #4 as the task complexity of App 1 and 2 in these cells is 10 times lower. This shows that the achievable revenue is to a large extent determined by the task complexities and the arrival intensities of the applications with high task complexity. On the contrary, the revenue is insensitive to the inter-arrival time distribution for given arrival rate (c.f. revenue in Cell #3 and Cell #4).



(a) Average revenue vs. number of arrivals for Cells #1 and #2



(b) Average revenue vs. number of arrivals for Cells #3 and #4

B. Operator's Policy

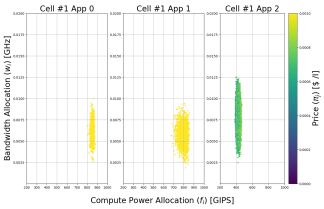
Fig. 2 shows the policy computed using ADRAP per application for cells $\#1, \#2, \#3, \#4$. The figure shows that the operator offers a higher price to WDs offloading an application with high computation and communication resource demands (c.f. App 0 and 1 in Cell $\#1$ and $\#2$). Recall that the threshold price $\overline{\pi}_i$ shown in Lemma 1 is an increasing function of d_i and L_{ϕ_i} which makes WDs willing to offload for higher prices. This shows that ADRAP can learn the reservation costs (C_i^0) of the WDs and adjusts the price accordingly. Similarly, the operator offers a high allocation of compute resources to WDs that have an application type with high compute resource demand to enable WDs to meet their task completion deadlines $\overline{\tau}_i^l$ (c.f. compare App 0 and 2 in Cell $\#1$). Interestingly, the operator allocates more communication resources to App 0 in Cell $\#2$ than in Cell $\#1$. This is due to that the arrival intensity of App 0 is higher in Cell #2 than in Cell #1, and the policy combines computation and communication resources to balance the offloading cost of the WDs. This also affects the allocations for App 2 for Cell $\#2$, as there is less communication resource left to be allocated to App 2, and hence a higher compute allocations are required. Similarly, we observe that when the arrival intensity is higher, as in Cell $\#3$ and Cell $\#4$, the operator allocates communication and computing resources more uniformly. This implies that communication resource allocation is to a large extent determined by the arrival intensity of the WDs.

C. Consumer Surplus and Offloading Probability

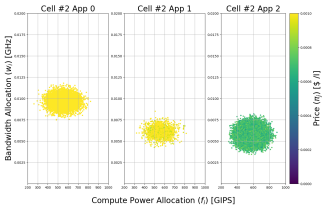
Fig. 3 shows the offloading probability and the average consumer surplus $(C_i^0 - C_i^1)$ as a function of the number of WDs that arrived. The figure shows that the policies computed by ADRAP and by PPO converge to similar values, showing the accuracy of the proposed approximation used in ADRAP. Observe that MROK has higher consumer surplus compared to ADRAP, which shows that it cannot adapt to the WDs' reservation costs, i.e., it offers a lower price. The figure also shows that MB-PPO cannot learn a good policy, owing to that the BNNs cannot approximate the state transition probabilities. The results for the offloading probability (Fig. 3a and Fig. 3b) confirm these observations, as a low price would result in a higher offloading probability.

Parameter	$f_o(.,.;\mathcal{W}^o)$	$f_\tau(.,.;\mathcal{W}^\tau)$
Hidden Layers	2	2
Number of Hidden Neurons	50	50
Hidden Layer Activation	ReLu	ReLu
Output Activation	$(tanh(5x)+1)/2$	Linear
N_d	50	50
Batch size	$N_{tra}/10$	$N_{tra}/10$
$\tilde{\sigma}^o, \tilde{\sigma}^\tau$	1	0.1

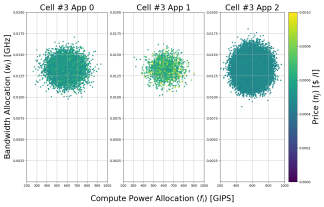
TABLE II: Hyperparameters for training the BNNs that estimate the offloading probability and the task completion time.



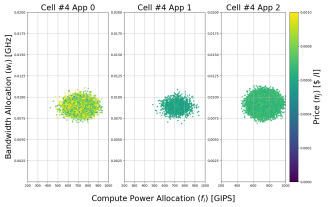
(a) Operator's policy per application for Cell $\#1$.



(b) Operator's policy per application for Cell $\#2$.



(c) Operator's policy per application for Cell $\#3$.



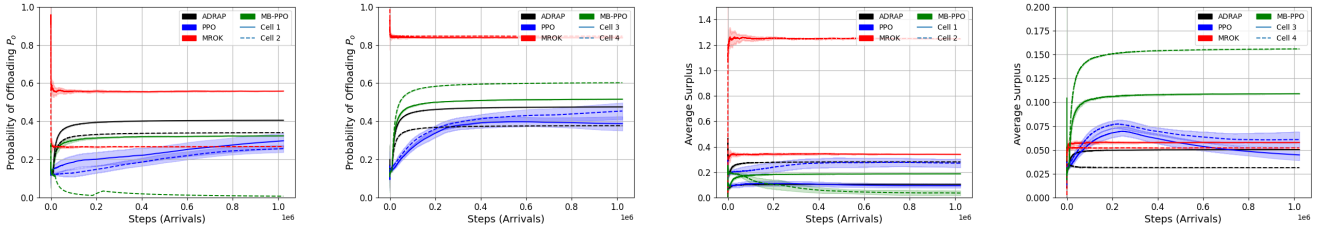
(d) Operator's policy per application for Cell $\#4$.

Fig. 2: Operator's policy per application for cells $\#1$ to $\#4$.

VII. Related Work

The joint optimization of edge compute and communication resources and pricing has been considered in a handful of recent works [4], [5], [20]. Authors in [4] proposed an auction for resource allocation and offloading, where resource allocation is based on bids from the WDs for a portion of the available edge resources. In [5], authors model the interaction of a revenu maximizing operator of EC and WDs as a Stackelberg game, and assume that WDs request the optimal amount of edge resources under a budget constraint. In [20], authors use a game theoretic approach to model the strategic interaction between users and the services. Unlike our work, calculating the optimal price and resource allocation require prior information about the traffic and workload characteristics and WDs' parameters, which could be unrealistic in practice. These assumptions are usually made to have an analytically tractable solution or a solution that could be computed efficiently in real time. Different from these works, our solution infers the dynamics of the traffic without assuming complete information.

Most related to ours are recent works that consider pricing under dynamic workload and incomplete information [6], [7], [11], [21], but these work are different from our work in two important ways. First, for the works [6], [7], the training phase is very long, which makes the proposed approaches



(a) Probability of offloading P_o vs. number of arrivals for cells $\#1$ and $\#2$.

(b) Probability of offloading P_o vs. number of arrivals for cells $\#3$ and $\#4$

(c) Average consumer surplus vs. number of arrivals for cells $\#1$ and $\#2$.

(d) Average consumer surplus vs. number of arrivals for cells $\#3$ and $\#4$.

Fig. 3: Offloading probability (P_o) and average consumer surplus vs. number of arrivals for cells $\#1$ to $\#4$, from left to right.

infeasible in practice, since training requires exploration of prices, which can lead to revenue loss for a long period of time. The pricing scheme proposed in [7] requires training on 10^4 servers, while the number of training epochs (i.e., days) is 400 in [6, Fig.1]. Second, authors in [11], [21] propose algorithms for fast computation of the optimal pricing under dynamic workload, these works do not consider resource optimization. Unlike these works, we propose an approach for increasing the sample efficiency through model-based learning, using BNNs to approximate the edge environment dynamics, and eliminate potential long-term revenue loss of the edge operator due to slow learning of the optimal policy.

VIII. Conclusion

We have considered the joint optimization of pricing and computing and communication resource allocation for task offloading in EC under a dynamic workload. We showed that the resulting sequential decision making problem can be modeled as a MDP, and we proposed a sample efficient model-based RL algorithm based on a novel approximation of the environment dynamics using BNNs approximators. We used simulations to show that our proposed approach can learn a good policy several orders of magnitude faster than model-free RL, and outperforms online learning in terms of the profit of the edge operator. Interesting directions of future work include exploring a multi-edge cell environment with diverse workload characteristics and WDs. Our approach could possibly be extended to transfer information among cells and would thus allow faster training and increased operator revenue.

References

[1] J. Yan, S. Bi, L. Duan, and Y.-J. A. Zhang, "Pricing-driven service caching and task offloading in mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4495–4512, 2021.

[2] G. Mitsis, E. E. Tsiropoulou, and S. Papavassiliou, "Price and risk awareness for data offloading decision-making in edge computing systems," *IEEE Systems Journal*, 2022.

[3] F. Tütüncüoğlu and G. Dán, "Optimal service caching and pricing in edge computing: a bayesian gaussian process bandit approach," *IEEE Transactions on Mobile Computing*, pp. 1–15, 2022.

[4] T. Bahreini, H. Badri, and D. Grosu, "Mechanisms for resource allocation and pricing in mobile edge computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 667–682, 2022.

[5] Y. Chen, Z. Li, B. Yang, K. Nai, and K. Li, "A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing," *Future Generation Computer Systems*, vol. 108, pp. 273–287, 2020.

[6] S. Chen, L. Li, Z. Chen, and S. Li, "Dynamic pricing for smart mobile edge computing: a reinforcement learning approach," *IEEE Wireless Communications Letters*, 2021.

[7] Z. Tang, F. Zhang, X. Zhou, W. Jia, and W. Zhao, "Pricing model for dynamic resource overbooking in edge computing," *IEEE Transactions on Cloud Computing*, 2022.

[8] S. Jošilo and G. Dán, "Joint management of wireless and computing resources for computation offloading in mobile edge clouds," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1507–1520, 2021.

[9] L. Tan, Z. Kuang, L. Zhao, and A. Liu, "Energy-efficient joint task offloading and resource allocation in ofdma-based collaborative edge computing," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1960–1972, 2022.

[10] S. Jošilo and G. Dán, "Joint wireless and edge computing resource management with dynamic network slice selection," *IEEE Transactions on Networking*, vol. 30, no. 4, pp. 1865–1878, 2022.

[11] Z. Zhang, Z. Li, and C. Wu, "Optimal posted prices for online cloud resource allocation," in *Proc. of the ACM on Measurement and Analysis of Computing Systems*, jun 2017, pp. 1–26.

[12] H. R. Varian, *Microeconomic analysis*, 3rd ed. Norton, New York, 1992.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[14] T. W. Killian, S. Daulton, G. Konidaris, and F. Doshi-Velez, "Robust and efficient transfer learning with hidden parameter markov decision processes," in *in Proc. of Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft, "Learning and policy search in stochastic dynamical systems with bayesian neural networks," in *in Proc. of Int. Conf. on Learning Representations*, 2017.

[16] J. Hernandez-Lobato, Y. Li, M. Rowland, T. Bui, D. Hernandez-Lobato, and R. Turner, "Black-box alpha divergence minimization," in *in Proc. of Int. Conf. on Machine Learning*, vol. 48, New York, New York, USA, Jun 2016, pp. 1511–1520.

[17] F. Tütüncüoğlu and G. Dán, "Joint resource management and pricing for task offloading in serverless edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–15, 2023.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[20] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 302–317, 2021.

[21] F. Lyu, X. Cai, F. Wu, H. Lu, S. Duan, and J. Ren, "Dynamic pricing scheme for edge computing services: A two-layer reinforcement learning approach," in *in Proc. of IEEE Int. Symposium on Quality of Service*, 2022.