Bayesian Optimization of Queuing-based Multi-Channel URLLC Scheduling

Wenheng Zhang, Mahsa Derakhshani, *Senior Member, IEEE*, Gan Zheng, *Fellow, IEEE*, Chung Shue Chen, *Senior Member, IEEE* and Sangarapillai Lambotharan, *Senior Member, IEEE*

***Abstract*—This paper studies the allocation of shared resources between ultra-reliable low-latency communication (URLLC) and enhanced mobile broadband (eMBB) in the emerging 5G and beyond cellular networks. In this paper, we design a unique queuing mechanism for the joint eMBB/URLLC system. The aim is to flexibly schedule URLLC traffic to enhance the total eMBB throughput and the reliability of URLLC packets (i.e., the probability of not dropping URLLC packets in each mini-slot) while maintaining a satisfactory transmission latency as per the 3GPP requirements. Precisely, by deriving the steady-state probabilities of URLLC queue backlog analytically, we formulate a stochastic optimization problem to maximize the total normalized eMBB throughput and the URLLC utility. Due to the stochastic nature of the objective function, it is expensive to evaluate it for any set of inputs, and thus the Bayesian optimization is applied to obtain the optimal results of such a black-box objective function. Numerical results demonstrate that the proposed queuing mechanism never violates the latency requirement of the URLLC services but improves the reliability. It also enhances the total normalized eMBB throughput as compared to the method without queuing.**

***Index Terms*—Bayesian optimization, dynamic scheduling, eMBB, punctured scheduling, queuing, URLLC.**

I. Introduction

THE fifth generation of cellular networks (5G) and the emerging sixth-generation wireless networks (6G) have to support extremely challenging service requirements, such as ultra low latency, ultra high reliability, and high transmission rates, which are essential to support emerging applications in the fields of autonomous driving, large-scale Internet of Things (IoT), and augmented reality/virtual reality (AR/VR) [1]–[5]. Ultra-reliable low-latency communications (URLLC) and enhanced Mobile Broadband (eMBB) services are two main categories that have been introduced in the 5G new radio. In the application of the former service, 3GPP has specified that each URLLC packets should be served in a very short time (i.e., up to one millisecond) with the packet error rate of less than 10^{-5}, or even down to 10^{-9} [6]. With such strict requirements, the possible applications of URLLC include intelligent transportation, industry automation and generally mission-critical services [7]. While URLLC has been characterized by low latency and ultra reliable requirements, the eMBB service targets high data rates to bring a faster and better user experience. The target data rates for eMBB services usually are higher than 100Mb/s [8].

When transmitting the eMBB and URLLC traffic in the same frequency-time spectrum, different time scales are applied to optimize the resource blocks allocation [9]. The eMBB services often occupy a more significant proportion of the transmission resource blocks, and the infrequent URLLC traffic is normally served spontaneously [10]. Thus, each eMBB packet can be coded over a longer time slot and each URLLC packet is served in shorter mini-slot to achieve the low latency requirement. Besides, 3GPP proposed a resource allocation and puncturing scheme for the URLLC/eMBB coexistence system [11]. In such puncturing scheme, the transmission channel suspends the ongoing eMBB packets in order to transmit URLLC packets. Specifically, when the channel senses an incoming URLLC packet, it immediately punctures the eMBB traffic in the next mini-slot, or the URLLC packet will be dropped as a loss. The authors in [12] elaborated the benefits of using the puncturing scheduling policy on the joint system of eMBB and URLLC, which are considering the sporadic and random characteristics of the URLLC traffic and transmitting such latency-critical service upon the ongoing eMBB transmissions to enhance the reliability. However, such a scheduling technique influences the quality of services of the eMBB services. In more detail, the utility of eMBB services (i.e., the data rate or total throughput) is directly influenced by the number of preemption URLLC packets and more puncturing on the eMBB channels could cause the decoding of eMBB to fail.

A. Relevant Works

With the preemptive puncturing policy, the current studies focus on the optimization problems that maximize the allocation of URLLC packets among eMBB traffic within the target latency while maintaining the utility of eMBB users.

Some researchers assumed the URLLC traffic has a higher scheduling priority than eMBB traffic. For example, the work in [13] considered a risk-sensitive model for eMBB transmission and optimized the URLLC reliability as a chance constraint which was relaxed based on Markov's inequality. Besides, in [14], the authors showed the advantage of eMBB-aware URLLC scheduling decisions. Such a method primarily punctures eMBB transmission, which caused the low modulation and coding scheme index. Some other researchers attempted to jointly schedule URLLC with eMBB services.

Wenheng Zhang, Mahsa Derakhshani, Gan Zheng and Sangarapillai Lambotharan are with Signal Processing and Networks Research Group, Wolfson School, Loughborough University, Loughborough LE11 3TU, U.K. (email: W.Zhang2@lboro.ac.uk; M.Derakhshani@lboro.ac.uk; G.Zheng@lboro.ac.uk; S.Lambotharan@lboro.ac.uk).

Chung Shue Chen is with Nokia Bell Labs, Paris-Saclay, Nozay 91620, France. (email: Chung_Shue.Chen@nokia-bell-labs.com)

In [15], the authors considered three different loss models for eMBB rate, i.e., linear, convex and threshold. They maximized the eMBB utility under different loss models while guaranteeing the URLLC requirements. Considering the different distributions of URLLC traffic, [16] formed a chance-constrained optimization problem of the joint eMBB/URLLC system, and the cumulative distribution function (CDF) was applied to transform the chance constraint into a linear constraint. Their proposed algorithm showed a fair allocation of resource blocks to eMBB users.

Some researchers implemented machine learning to solve the optimization of the joint eMBB /URLLC system. For instance, in [17], the authors used a flexible transmission interval selection policy and applied supervised learning to the joint optimization. However, their studies scheduled eMBB and URLLC unevenly (i.e., scheduled URLLC in advance). The authors in [18] proposed a long-term monitor system under the changing environment, which balanced the cost of eMBB traffic and the gain of URLLC services. Remarkably, the previous research has also shown that deep reinforcement learning (DRL) has significant advantages in enhancing the eMBB requirements in such a coexistence system. E.g., In [8], the authors proposed a model-free DRL to minimize the impact of puncturing on eMBB users during the URLLC transmission. The deep deterministic policy gradient based method was used in their system to solve the continuous action domain for the resource scheduling. Besides, [19] reduced the impact of URLLC immediate scheduling policy on eMBB utility while considering the variance of eMBB data rate. The DRL-based model intelligently assigned the URLLC packets among the eMBB users.

The existing studies have already achieved significant outcomes on the design of URLLC scheduling policy when it coexists with eMBB traffic, e.g., [13]–[18]. However, in general, all of the works mentioned above assumed that the incoming URLLC packets could only either be served or dropped immediately at the beginning of each mini-slot, which only ensured the URLLC latency requirement but caused packet losses to the URLLC. The work in [20] considered a queuing mechanism for URLLC traffic, where each transmission channel had an individual queue. It is assumed that the packet arrivals are evenly distributed between different queues. The base station (BS) makes the puncturing decision of each eMBB channel separately in turns at the beginning of each time slot. The even distribution of packet arrivals and separate decision making of different channels would lead to more drops depending on the channel conditions. Therefore, in this paper we propose a new queuing scheme with a single queue wherein the dynamic scheduling of the multi-channel system could be managed jointly and hence more effectively. We have implemented the results of [20] as a benchmark and verified the effectiveness of the new scheduling scheme as compared with [20].

B. Motivation and Contributions

In this paper, we propose a novel queue-based allocation mechanism for the URLLC traffic in a multi-channel system and aim to improve the URLLC reliability within the target latency by simultaneously queuing/transmitting multiple URLLC packets in each mini-slot over multiple channels. The main contributions of this paper are summarized as follows:

- We study the URLLC scheduling in a coexisting URLLC/eMBB downlink transmission system. In such a joint system, we assume the BS has already allocated the radio resources among the eMBB users on different channels at the beginning of each eMBB time slot before scheduling the URLLC packets. We propose a M/G/∞ queuing mechanism for infrequent URLLC traffic and derive the expressions that analyze the URLLC queuing mechanism, including the number of queuing packets and the instantaneous mean latency of the served URLLC packets. The differences between the analytical and numerical results are below 10^{-3}. The results also show that the proposed queuing mechanism can wisely adjust the number of URLLC queuing packets to balance the tradeoff between the reliability and latency requirements of URLLC services.
- We apply the puncturing scheduling for the joint eMBB/URLLC system and propose an objective function that maximizes the total normalized eMBB throughput and the URLLC utility among all possible system states. The URLLC utility is quantified as the successful transmission probability of URLLC traffic with no drop over all channels which considers the probabilities of successful transmission of URLLC traffic on each channel and the probability of not dropping URLLC packets. However, it is complicated to derive the closed-form of the steady state probabilities by analyzing the dynamic of the URLLC queue. The standard optimization techniques cannot be used in the case of solving a black-box objective function, thus, in this paper, we apply the Bayesian optimization (BO) algorithm to solve the optimization problem. Besides, we use the additive structure with the BO algorithm to reduce high dimensions by decomposing the original search space into several sub-spaces. We implement different acquisition functions of the BO algorithm and use the random search as a comparison to show the better performance of BO. The results show that the BO algorithm can quickly reach the optimal point with a limited training set.
- We evaluate the performance of the proposed algorithms by comparing them with two benchmarks. One is the optimization problem for the case of without the queuing scheme, another is the proposed mechanism in [20]. Based on the results, firstly, our one queue scheme shows the flexibility of the URLLC scheduling compared to [20], where the scheduling of the URLLC is rigid. Beside, the scheme without queuing is more sensitive to the total number of channels and the value of URLLC arrival loads. Without enough channels, the mean value of total eMBB throughput of the no queuing mechanism is significantly lower than the queue-based scheme. Moreover, we define the URLLC outage probability as the probability that there exists dropped URLLC packets in a mini-

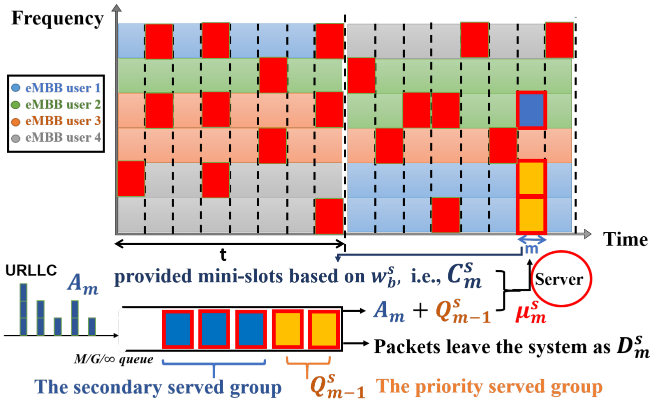
Frequency
eMBB user 1
eMBB user 2
eMBB user 3
eMBB user 4
Time
t
m
URLLC
A_m
provided mini-slots based on w_b^s, i.e., C_m^s
Server
$A_m + Q_{m-1}^s$
μ_m^s
Packets leave the system as D_m^s
M/G/∞ queue
The secondary served group
Q_{m-1}^s
The priority served group

Figure 1. Scheduling policy for URLLC in the joint eMBB/URLLC system with the proposed $M/G/\infty$ queuing mechanism. The served URLLC packets are shown in red rectangles at each mini-slot. New arriving and queue backlog packets in mini-slot m are shown in black and yellow colours, respectively.

slot. A slight increase in the number of transmission channels can significantly improve the URLLC outage probability of the proposed queuing scheme. However, the URLLC outage probability of the scheme without queuing hardly meets the 3GPP requirements, even with a large number of channels. These results show that our proposed queuing system can enhance both the total normalized eMBB throughput and URLLC requirements.

C. Organisation

The rest of this paper is organized as follows. Section II introduces the multi-channel scheduling policy for URLLC with queuing in coexistence with eMBB. Moreover, the working principles of the URLLC queuing mechanism are explained and an optimization problem is formulated to find the optimal puncturing weights. In Section III, the dynamics of the proposed queuing are mathematically analyzed aiming to characterize the achievable performance of URLLC and eMBB traffic depending on the puncturing weights. Section IV reviews the BO framework with additive structures and focuses on developing an algorithm based on BO to optimize the URLLC scheduling. The detail of using the additive structure algorithm to reduce the high-dimension problem into a serial of low-dimensional sub-problems is also explained. Section V presents the numerical results to evaluate the performance of the queuing mechanism and the developed algorithm for URLLC scheduling. Finally, the conclusions are given in Section VI.

Notations: We use $\mathbb{E}[.]$ and $\mathbb{P}[.]$ to denote the expectation and the probability respectively. $\mathbb{N}$ and $\mathbb{N}_+$ are represented as the set of natural numbers and positive integers, respectively.

II. System Model and Problem Formulation

A. System Model

We consider a downlink wireless system that contains a base station (BS) to simultaneously provide eMBB and URLLC services. The system frequency resources are divided into B channels of equal bandwidth. We study the URLLC scheduling over a set of channels $\mathcal{B} = \{1, 2, \ldots, B\}$ in coexistence with eMBB traffic. These channels are primarily scheduled for eMBB transmission and will be shared with URLLC traffic adopting a puncturing policy as proposed in [15]. The puncturing policy will be optimally designed to simultaneously minimize the eMBB throughput loss due to URLLC transmission and enhance the URLLC reliability measure. For satisfying the URLLC latency requirement, it is assumed that the system operates on two different timescales as proposed by 3GPP [21], i.e., the eMBB time slot t, at the beginning of which the eMBB users are scheduled, and mini-slot m to schedule the sporadic URLLC packets. Each of the eMBB time slot is divided into M equal size mini-slots. The set of mini-slots is represented by $\mathcal{M} = \{1, 2, \ldots, M\}$. The frequency-time spectrum of the downlink transmission system on different scheduling intervals is illustrated in Fig. 1 along with Table I listing the notations used in this work.

Table I
Summary of Notations

Notation	Definition
$\mathcal{B}$, $\mathcal{M}$	Set of channels and mini-slots, respectively.
$\mathcal{C}_{embb}$	Set of possible channel rates for an eMBB user.
$\mathcal{C}_{urllc}$	Set of probability of successful transmission for URLLC.
m, t	URLLC time slot and eMBB time slot, respectively.
s^b,s^b_{urllc}	The eMBB channel rate and probability of successful transmission for URLLC at channel $b \in \mathcal{B}$, respectively.
$\mathcal{S}$, P_s	Set of possible system state and its PMF, respectively.
$\boldsymbol{W}$	Matrix of URLLC puncturing weights.
π	Bernoulli generate variable for URLLC puncturing.
A_m, $C_m^{\boldsymbol{s}}$	Number of URLLC packets that arrive and mini-slots allocated to URLLC in $m \in \mathcal{M}$, $\boldsymbol{s} \in \mathcal{S}$, respectively.
$Q_{m-1}^{\boldsymbol{s}}$	Number of URLLC packets already stored in the queue before $m \in \mathcal{M}$, and in $\boldsymbol{s} \in \mathcal{S}$.
$D_m^{\boldsymbol{s}}$, $\mu_m^{\boldsymbol{s}}$	Number of dropped and served URLLC packets in $m \in \mathcal{M}$ and $\boldsymbol{s} \in \mathcal{S}$, respectively.
p_a, $p_Q^{\boldsymbol{s}}$, $p_c^{\boldsymbol{s}}$	The PMF of A_m, $Q_{m-1}^{\boldsymbol{s}}$ and $C_m^{\boldsymbol{s}}$, respectively.
$U_{\text{embb}}^{\boldsymbol{s}}$	Total normalized eMBB throughput in $\boldsymbol{s} \in \mathcal{S}$.
$U_{\text{urllc}}^{\boldsymbol{s}}$	URLLC utility in $\boldsymbol{s} \in \mathcal{S}$.
$\boldsymbol{\Psi}^{\boldsymbol{s}}$	State transition probability of URLLC queue in $\boldsymbol{s} \in \mathcal{S}$.
$f(.)$, $f_{\boldsymbol{s}}(.)$	Black-box objective and its sub-objective function.
N_s	Number of subspace/combinations for S.
I	Decomposed input space
d	Input space of black-box function.
$\mathcal{N}(r)$	Training data set of proposed solution with r data pairs.
$\rho_0(.)$, $K_0(.)$	Prior mean function and covariance matrix, respectively.
$\rho_r(.)$, $\sigma_r^2(.)$	Posterior mean and variance, respectively.
$g(.)$	Acquisition function.
ξ	Trade-off parameter between exploitation and exploration.
R	Maximum number of iterations of proposed solution.
λ	URLLC arrival rate.
P_{out}	URLLC outage probability.
α	Mean latency of the served URLLC packets.
β	Number of the served URLLC have been queued.

We assume block fading for eMBB users where the channel state of an eMBB user at one frequency channel remains constant during one time slot. The set of possible channel rates for an eMBB user is represented by $\mathcal{C}_{embb} = \{\mathcal{C}_{embb,1}, \mathcal{C}_{embb,2}, \ldots, \mathcal{C}_{embb,N_c^{embb}}\}$, where N_c^{embb} is the cardinality of the set $\mathcal{C}_{embb}$. Let $\mathcal{S}_{embb} = [s^b]$ be the $1 \times B$ vector of the system state of eMBB that represents the instantaneous channel rates of eMBB users over B channels at current time slot, where $s^b \in \mathcal{C}_{embb}$ denotes the channel rate of the eMBB user transmitting at channel $b \in \mathcal{B}$. Moreover, we consider the probability of successful transmission as the system state of URLLC. The set of possible probabilities of successful transmission for URLLC wireless channels is denoted by $\mathcal{C}_{\text{urllc}} = \{\mathcal{C}_{urllc,1}, \mathcal{C}_{urllc,2}, \dots, \mathcal{C}_{urllc,N_c^{urllc}}\}$, where N_c^{urllc} is the cardinality of the set $\mathcal{C}_{urllc}$. Let $S_{urllc} = [s^b_{urllc}]$ be the $1 \times B$ vector to denote the probabilities of successful transmission over all channels at current time slot. $s^b_{urllc} \in \mathcal{C}_{urllc}$ represents probability of successful transmission of URLLC traffic on channel $b \in \mathcal{B}$ at current time slot. We use $S = [S_{urllc}, S_{embb}]$ to denote the system state of these two kinds of traffic; there are $N_s = (N_c^{urllc} N_c^{embb})^B$ combinations for S and the set of all possible combinations is denoted as $\mathcal{S} = \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_s}\}$. The probability mass function (PMF) of S is represented by $\mathrm{P}_s(\boldsymbol{s}) = \mathbb{P}(S = \boldsymbol{s})$, $\mathrm{P}_s(\boldsymbol{s}) \in [0,1]$, $\forall \boldsymbol{s} \in \mathcal{S}$. The wireless system undergoes the channel variations in each eMBB slot which are modeled as an independent and identically distributed random process over the set of $\mathcal{C}_{eMBB}$ and $\mathcal{C}_{urllc}$. Thus, our scheduler is assumed to know (or able to estimate over time) the distribution of system state (i.e., $\mathrm{P}_s(\boldsymbol{s})$) across the eMBB slot.

The URLLC transmissions are determined and realized by the puncturing weights selected in different channels given $\boldsymbol{s}$. Suppose $\boldsymbol{W} = \left[w^{\boldsymbol{s}}_b\right]$, $\forall b \in \mathcal{B}$, $\forall \boldsymbol{s} \in \mathcal{S}$ is the $B \times N_s$ matrix of URLLC puncturing weights, where $w^{\boldsymbol{s}}_b \in [0,1]$ is the weight of puncturing on the channel b in system state $\boldsymbol{s}$. We assume that the puncturing weights are the same over different mini-slots of one time-slot and hence time-homogeneous over one time-slot. The realization of each puncturing weight is random based on the variable $\pi^{\boldsymbol{s}}_{b,m}$, which is generated using a Bernoulli distribution with the success probability of $w^{\boldsymbol{s}}_b$ at the beginning of each mini-slot. If $\pi^{\boldsymbol{s}}_{b,m} = 1$, a URLLC packet will be overridden upon the scheduled eMBB transmission on channel b in mini-slot m when the system state is $\boldsymbol{s}$. We assume that each channel can transmit at most one URLLC packet per mini-slot. Otherwise, $\pi^{\boldsymbol{s}}_{b,m} = 0$ and channel b will not be punctured by any URLLC packet in mini-slot m when the system state is $\boldsymbol{s}$. Given $\pi^{\boldsymbol{s}}_{b,m}$, the total number of mini resource blocks allocated to URLLC puncturing over all channels in mini-slot m can be computed as

$$C^{\boldsymbol{s}}_m = \sum_{b\in\mathcal{B}} \pi^{\boldsymbol{s}}_{b,m}. \tag{1}$$

B. URLLC Scheduling Policy with Queuing

We propose the multi-channel scheduling for the URLLC services with one $M/G/\infty$ queue, which can simultaneously serve multiple URLLC packets at the beginning of each mini-slot to be punctured over different channels depending on $\pi^{\boldsymbol{s}}_{b,m}$. In particular, M means we consider Poisson URLLC arrivals, G means that service times have a General distribution, and ∞ specifies that the queue length is considered infinite. As supported by [21], we adopt Poisson URLLC packet arrivals that enable us to analytically derive the expected number of served URLLC packet and the probability of no drop. Moreover, we consider infinite queue length in this work so that we could analyze the number of URLLC packet drops that caused by lack of radio resources as well as complying with delay requirements and not limited by the length of the queue.

The proposed queuing mechanism enhances the URLLC reliability by not immediately dropping unserved packets and keep them in a queue while meeting the strict latency requirements. 3GPP has suggested an extreme latency requirement for URLLC, i.e., each URLLC packet should be served within 0.3 milliseconds [11], [21], [22]. In addition, 3GPP advises that the length of each URLLC mini-slot is 0.125 milliseconds. Accordingly, we establish the working principle of URLLC queuing mechanism as the URLLC packets can wait in the queue until next mini-slot if it cannot be transmitted immediately. If a URLLC packet has already queued for one mini-slot and will not be transmitted in the next mini-slot, it will be dropped from the queue and counted as a loss.

We assume that URLLC packets have the same fixed packet size and randomly arrive in each mini-slot. Let A_m denote the number of URLLC packets that arrive at the queue line between mini-slot $m-1$ and mini-slot m.

Suppose PMF of A_m is denoted by $p_a(y)$ as

$$p_a(y) = \mathbb{P}(A_m = y), \forall y \in \mathbb{N}, \tag{2}$$

where $p_a(y) \in [0,1]$ and $\sum_{y\in\mathbb{N}} p_a(y) = 1$. Let $\lambda = \mathbb{E}\left[A_m\right]$ denote the average URLLC arrival rate (in units of packets/mini-slot).

Let $Q^{\boldsymbol{s}}_{m-1}$ be the number of URLLC packets already stored in the queue before the mini-slot m in system state $\boldsymbol{s}$. Thus, based on the working principle of the queuing mechanism, if in mini-slot m system state $\boldsymbol{s}$, the number of served URLLC packets $\mu^{\boldsymbol{s}}_m$ and the number of dropped URLLC packets $D^{\boldsymbol{s}}_m$ can be derived as

$$\mu^{\boldsymbol{s}}_m = \min\left(C^{\boldsymbol{s}}_m, Q^{\boldsymbol{s}}_{m-1} + A_m\right), \tag{3}$$
$$D^{\boldsymbol{s}}_m = \max\left(Q^{\boldsymbol{s}}_{m-1} - \mu^{\boldsymbol{s}}_m, 0\right). \tag{4}$$

Accordingly, in system state $\boldsymbol{s} \in \mathcal{S}$, the one-step queuing equation for the queue link is expressed as

$$Q^{\boldsymbol{s}}_m = Q^{\boldsymbol{s}}_{m-1} + A_m - \left(\mu^{\boldsymbol{s}}_m + D^{\boldsymbol{s}}_m\right), \forall m \in \mathcal{M}. \tag{5}$$

Fig. 2 illustrates an example on how the URLLC queuing mechanism operates in mini-slots. Suppose in this case that the number of provided channels is not sufficient to transmit the URLLC backlog packets in mini-slot m, i.e., $C^{\boldsymbol{s}}_m < Q^{\boldsymbol{s}}_{m-1}$. The reserved queue backlog will be dropped and the newly arrived packets will be backlogged in the queue backlog, i.e., $Q^{\boldsymbol{s}}_m = A_m$. In the second case, when the system provides enough transmission opportunities for URLLC queue backlogs and there are some surplus opportunities for the newly arrived packets, there will be no drops and the queue backlog in the next mini-slot becomes the reserve of newly arrived packets.

C. Problem Formulation

Here, we formulate the URLLC multi-channel scheduling problem as a stochastic optimization problem that optimally determines the puncturing weights for all channels and all system states. Since the URLLC packets will be punctured on the ongoing eMBB transmissions, the URLLC scheduling policy should strike the balance between mitigating the adverse

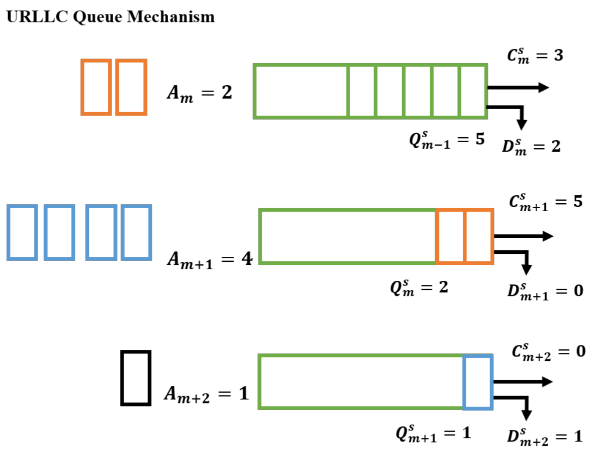

Figure 2. Illustration of the working principle of URLLC queuing mechanism.

impact of URLLC puncturing on the average eMBB throughput and satisfying the URLLC requirements. Therefore, we formulate an optimization problem aiming to maximize the total normalized eMBB throughput and URLLC utility. The URLLC utility is quantified as the successful transmission probability of URLLC traffic with no drop over all channels in the system state $\boldsymbol{s} \in \mathcal{S}$.

Accordingly, the optimization problem is formulated as:

$$\max_{\boldsymbol{W}} \quad \sum_{\mathbf{s}\in\mathcal{S}} \mathrm{P}_s(\mathbf{s}) \left(U^{\boldsymbol{s}}_{\text{embb}} + U^{\boldsymbol{s}}_{\text{urllc}}\right), \tag{6a}$$

$$\textbf{s.t.,} \quad w^{\boldsymbol{s}}_b \in [0,1], \forall b \in \mathcal{B}, \forall \mathbf{s} \in \mathcal{S}, \tag{6b}$$

where $U^{\boldsymbol{s}}_{\text{embb}}$ denotes the total normalized eMBB throughput and $U^{\boldsymbol{s}}_{\text{urllc}}$ represents URLLC utility over all channels in system state $\boldsymbol{s} \in \mathcal{S}$. The constraint in (6b) ensures that the value of puncturing weights could only vary between zero and one.

The URLLC utility $U^{\boldsymbol{s}}_{\text{urllc}}$ is defined as the product of the probability that no URLLC packet would be dropped in each mini-slot (i.e., $\mathbb{P}(D^{\boldsymbol{s}}_m = 0)$) and the successful transmission probability of URLLC traffic over all channels in system state $\boldsymbol{s} \in \mathcal{S}$. Thus, $U^{\boldsymbol{s}}_{\text{urllc}}$ is expressed as

$$U^{\boldsymbol{s}}_{\text{urllc}} = \mathbb{P}(D^{\boldsymbol{s}}_m = 0) \sum_{b\in\mathcal{B}} s^b_{urllc} \frac{w^{\boldsymbol{s}}_b}{\sum_{b'\in\mathcal{B}} w^{\boldsymbol{s}}_{b'}}. \tag{7}$$

The total eMBB throughput in system state $\boldsymbol{s} \in \mathcal{S}$ is defined as the total eMBB rate over all channels (i.e., $\sum_{b\in\mathcal{B}} s^b$) scaled by $(1 - w^{\boldsymbol{s}}_b \mathbb{E}\left[\mu^{\mathrm{s}}_m\right] / \sum_{b\in\mathcal{B}} w^{\boldsymbol{s}}_b)$ that is the ratio of one time slot the eMBB used excluding URLLC transmissions. In particular, $w^{\boldsymbol{s}}_b \mathbb{E}\left[\mu^{\mathrm{s}}_m\right] / \sum_{b\in\mathcal{B}} w^{\boldsymbol{s}}_b$ represents the expected number of mini-slots punctured for URLLC service on channel b to the total number of mini-slots in one eMBB slot in system state $\boldsymbol{s} \in \mathcal{S}$.

Therefore, the total normalized eMBB throughput in system state $\boldsymbol{s} \in \mathcal{S}$ can be calculated as

$$U^{\boldsymbol{s}}_{\text{embb}} = \frac{\sum_{b\in\mathcal{B}} s^b \left(1 - \frac{w^{\boldsymbol{s}}_b}{\sum_{b\in\mathcal{B}} w^{\boldsymbol{s}}_b} \mathbb{E}\left[\mu^{\mathrm{s}}_m\right]\right)}{\sum_{b\in\mathcal{B}} s^b}. \tag{8}$$

By solving the optimization problem (6) and obtaining the optimal puncturing weights, the proposed scheduling policy enhances the URLLC reliability by maximizing $U^{\boldsymbol{s}}_{\text{urllc}}$ in (6), while the URLLC latency requirement is supported by the URLLC queuing mechanism. To solve (6), we need to derive $\mathbb{E}\left[\mu^{\boldsymbol{s}}_m\right]$ and $\mathbb{P}(D^{\boldsymbol{s}}_m = 0)$ by studying the queue dynamics. The next section provides the queuing analysis to establish the objective function as the function of $\boldsymbol{W}$.

III. Performance Analysis with Queuing

The optimization equation in (6) contains two metrics that demonstrate the efficacy of the URLLC scheduling policy when coexisting with eMBB services, i.e., the total normalized eMBB throughput and URLLC utility. In this section, we mathematically analyze the URLLC queuing mechanism in order to derive the objective function as a function of $\boldsymbol{W}$.

A. *eMBB Throughput Analysis*

Here, we start the analysis by studying the eMBB throughput in (8). This requires the analysis of $\mathbb{E}\left[\mu^{\mathrm{s}}_m\right]$. Let $p^{\boldsymbol{s},m}_{\mu}(x)$ denote $\mathbb{P}\big(\mu^{\boldsymbol{s}}_m = x\big)$ where x represents the number of served URLLC packets, $x \in \{0, \ldots, B\}$. Accordingly, the expected number of served URLLC packets in mini-slot m system state $\boldsymbol{s} \in \mathcal{S}$ can be expressed as

$$\mathbb{E}\big[\mu^{\mathbf{s}}_m\big] = \sum_{x=0}^{B} x \cdot p^{\boldsymbol{s},m}_{\mu}(x). \tag{9}$$

Therefore, the first step to find the expected value of served URLLC packets is deriving the PMF of served URLLC packets, i.e., $p^{\boldsymbol{s},m}_{\mu}(x)$. Considering (3), the event of $\big(\mu^{\boldsymbol{s}}_m = x\big)$ occurs when either $(C^{\boldsymbol{s}}_m = x)$ or $(Q^{\boldsymbol{s}}_{m-1} + A_m = x)$. Consequently,

$$\begin{aligned} p^{\boldsymbol{s}}_{\mu}(x) = {} & \mathbb{P}(C^{\boldsymbol{s}}_m = x)\mathbb{P}(Q^{\boldsymbol{s}}_{m-1} + A_m \geq x) \\ & + \mathbb{P}(C^{\boldsymbol{s}}_m > x)\mathbb{P}(Q^{\boldsymbol{s}}_{m-1} + A_m = x), \forall x \in \mathcal{B}. \end{aligned} \tag{10}$$

Therefore, to derive $p^{\boldsymbol{s}}_{\mu}(x)$, we need to calculate $p^{\boldsymbol{s}}_c(x) = \mathbb{P}(C^{\boldsymbol{s}}_m = x)$ and $\mathbb{P}(A_m + Q^{\boldsymbol{s}}_{m-1} = x)$.

Let us start with the derivation of $p^{\boldsymbol{s}}_c(x)$ based on (1). Considering that the event $(\pi^{\boldsymbol{s}}_{b,m} = 1)$ happens with the probability of $w^{\boldsymbol{s}}_b$, $p^{\boldsymbol{s}}_c(x)$ can be derived as a function of puncturing weights as

$$p^{\boldsymbol{s}}_c(x) = \sum_{\mathcal{K}_C\in\mathcal{F}_C} \prod_{\delta\in\mathcal{K}_C} w^{\boldsymbol{s}}_{\delta} \prod_{\delta'\in\mathcal{K}_C{}'} (1 - w^{\boldsymbol{s}}_{\delta'}), \tag{11}$$

where $\mathcal{F}_C$ contains all the combinations that choose x channels out of $\mathcal{B}$. Also, $\mathcal{K}_C$ and $\mathcal{K}_C{}'$ denote possible subsets of $\mathcal{F}_C$ and its complementary set, respectively. Moreover, $\delta \in \mathcal{K}_C$ denotes the channels that are chosen to transmit URLLC packets and $\delta^{'} \in \mathcal{K}_C{}'$ denotes the channels in which no URLLC puncturing happens. According to (11), note that $p^{\boldsymbol{s}}_c$ is *time-homogeneous* across mini-slots in one eMBB slot and does not depend on m. This is because the puncturing weights are also time-homogeneous.

Moreover, the probability of $\mathbb{P}(Q^{\boldsymbol{s}}_{m-1} + A_m = x)$ can be written as

$$\mathbb{P}(A_m + Q^{\boldsymbol{s}}_{m-1} = x) = \sum_{y=0}^{x} p_a(y) \cdot p^{\boldsymbol{s}}_Q(x - y), \tag{12}$$

where $p_a(y) = \mathbb{P}(A_m = y)$ is the PMF that the number of the newly arrived URLLC packets is $A_m = y$ as introduced in (2) and $p_Q^{\boldsymbol{s}}(x-y)$ represents the steady state probability of the URLLC queuing backlog. In particular,

$$p_Q^{\boldsymbol{s}}(x-y) = \mathbb{P}(Q_{m-1}^{\boldsymbol{s}} = x-y). \tag{13}$$

Based on (12), the next step of deriving $\mathbb{E}\big[\mu_m^{\mathrm{s}}\big]$ in order to obtain the total normalized eMBB throughput is to find the steady state probability of the URLLC queue backlog.

B. Steady State Probability of the URLLC Queuing Mechanism

To obtain the steady state probabilities of the URLLC queue backlog, we apply the Markov chain model considering the transition probabilities among them [23]–[25]. In particular, we assume that the state of the Markov chain represents the number of URLLC packets backlogged in the queue that can change from zero to infinity. State zero represents the empty queue. The state transition probability matrix is denoted by $\boldsymbol{\Psi}^{\boldsymbol{s}} = [\psi_{n,k}^{\boldsymbol{s}}]$ where

$$\psi_{n,k}^{\boldsymbol{s}} = \mathbb{P}(Q_m^{\boldsymbol{s}} = k | Q_{m-1}^{\boldsymbol{s}} = n), \forall n, k \in \mathbb{N}, \tag{14}$$

and,

$$\sum_{k=0}^{\infty} \psi_{n,k}^{\boldsymbol{s}} = 1. \tag{15}$$

The steady-state solution of the Markov chain in system state $\boldsymbol{s} \in \mathcal{S}$, denoted by $p_Q^{\boldsymbol{s}}(q)$, $q \in \mathbb{N}$, can be calculated as [23]

$$p_Q^{\boldsymbol{s}}(q) = F(\mathbf{0}_{q,q})[F(\boldsymbol{\Psi}^{\boldsymbol{s}} - \mathrm{I})]^{-1}, \tag{16}$$

where the output of the function $F(\mathbf{A})$ is the matrix $\mathbf{A}$ with its right-most column replaced with all 1's. I is the identity matrix of size q, $\mathbf{0}_{q,q}$ denotes the zero matrix of size $q \times q$.

***Proposition** 1: The time-homogeneous transition probabilities of the URLLC queuing mechanism given $\boldsymbol{s} \in \mathcal{S}$ over eMBB slot can be computed as:*

$$\psi_{n,k}^{\boldsymbol{s}} = \begin{cases} p_a(k) \displaystyle\sum_{x=0}^{\min\{B,n\}} p_c^{\boldsymbol{s}}(x) + \sum_{\theta=1}^{B-n} \Big(\sum_{y=k}^{\theta} p_a(y) \Big) p_c^{\boldsymbol{s}}(n+\theta), \\ \text{if } k = 0, \quad \forall n \in \mathbb{N}, \\ p_a(k) \displaystyle\sum_{x=0}^{\min\{B,n\}} p_c^{\boldsymbol{s}}(x) + \sum_{\theta=1}^{B-n} p_a(k+\theta) p_c^{\boldsymbol{s}}(n+\theta), \\ \text{if } \forall k \in \mathbb{N}_+, \quad \forall n \in \mathbb{N}. \end{cases} \tag{17}$$

Proof: Given $\boldsymbol{s} \in \mathcal{S}$, the steps of finding the transition probabilities $\psi_{n,k}^{\boldsymbol{s}}$, $\forall n, k \in \mathbb{N}$ can be divided into two cases as follows:

Case 1: In this case, the newly arrived packets cannot be transmitted in the current mini-slot and need to be backlogged in the queue. Hence, $Q_m^{\boldsymbol{s}} = A_m$.

Table II
THE ANALYSIS OF THE URLLC QUEUING MECHANISM IN MINI-SLOT $m \in \mathcal{M}$ GIVEN $\boldsymbol{s} \in \mathcal{S}$.

Provided mini-slots $C_m^{\boldsymbol{s}}$	**Served packets** $\mu_m^{\boldsymbol{s}}$	**Drops** $D_m^{\boldsymbol{s}}$	**Updated backlog** $Q_m^{\boldsymbol{s}}$
Part (a): The Parameters of Queue when $Q_{m-1}^{\boldsymbol{s}} = n > B$			
0	0	n	A_m
1	1	n-1	A_m
⋮	⋮	⋮	⋮
B	B	n-B	A_m
Part (b):The Parameters of Queue when $Q_{m-1}^{\boldsymbol{s}} = n \leq B$			
0	0	n	A_m
1	1	n-1	A_m
⋮	⋮	⋮	⋮
n	n	0	A_m
n+1	n+min(A_m,1)	0	A_m-min(A_m,1)
n+2	n+min(A_m,2)	0	A_m-min(A_m,2)
⋮	⋮	⋮	⋮
n+(B-n)	n+min(A_m,B-n)	0	A_m-min(A_m,B-n)

There are two scenarios in which Case 1 could occur. First, the number of transmission channels is not enough (i.e., $B < Q_{m-1}^{\boldsymbol{s}}$ as shown in Table. II part (a)). In this scenario, the number of provided mini-slots for URLLC packets is always less than the number of queue backlog (i.e., $C_m^{\boldsymbol{s}} < Q_{m-1}^{\boldsymbol{s}}$ in mini-slot m). As a result, URLLC packets are bound to be lost, so the number of dropped URLLC packets is $D_m^{\boldsymbol{s}} = Q_{m-1}^{\boldsymbol{s}} - \mu_m^{\boldsymbol{s}} = Q_{m-1}^{\boldsymbol{s}} - C_m^{\boldsymbol{s}} > 0$. Consequently, the newly arrived URLLC packets will be queued and hence $Q_m^{\boldsymbol{s}} = A_m$. Therefore, the transition probability of such a scenario in system state $\boldsymbol{s} \in \mathcal{S}$ can be calculated as

$$\psi_{n,k}^{\boldsymbol{s}} = p_a(k) \sum_{x=0}^{B} p_c^{\boldsymbol{s}}(x), \forall k \in \mathbb{N}, n > B. \tag{18}$$

The second scenario of Case 1 is when $B > Q_{m-1}^{\boldsymbol{s}}$, but the number of provided resource blocks is no larger than the number of queuing packets (i.e., $C_m^{\boldsymbol{s}} \leq Q_{m-1}^{\boldsymbol{s}} \leq B$ as shown in Table. II part (b)). In this scenario, the next queue state only depends on the value of A_m. The transition probabilities of such a scenario can be calculated as

$$\psi_{n,k}^{\boldsymbol{s}} = p_a(k) \sum_{x=0}^{n} p_c^{\boldsymbol{s}}(x), \forall k \in \mathbb{N}, 0 \leq n \leq B. \tag{19}$$

To sum up, the transition probabilities of the two scenarios for Case 1 can be combined as the first term in (17), i.e., $p_a(k) \sum_{x=0}^{\min\{B,n\}} p_c^{\boldsymbol{s}}(x)$, for any $k \in \mathbb{N}$.

In Case 2, the number of provided mini-slots is larger than the number of URLLC backlog packets, and the BS starts to allocate mini-slots for transmitting the newly arrived packets, i.e., $Q_{m-1}^{\boldsymbol{s}} < C_m^{\boldsymbol{s}} \leq B$ as shown in Table. II gray parts. Let θ, $\theta \in \{1, \ldots, B-n\}$ denote the number of provided mini-slots for transmitting the newly arrived URLLC packets. In this case, the number of provided mini-slots is expressed as $C_m^{\boldsymbol{s}} = n + \theta, C_m^{\boldsymbol{s}} \in \{n+1, \ldots, B\}$. In Case 2, the queuing mechanism will not drop any URLLC packets from the queue backlog (i.e., $D_m^{\boldsymbol{s}} = 0$). If the number of newly arrived URLLC packets is larger than θ, the remnant URLLC packets will be queued in the next mini-slot, i.e., $Q^{\boldsymbol{s}}_m = A_m - \theta > 0$, $k \in \mathbb{N}_+$. Otherwise, all the new arriving URLLC packets will be transmitted and the next state goes to state zero, i.e., $A_m \leq \theta$, $Q^{\boldsymbol{s}}_m = A_m - A_m = 0$, $k = 0$. The transmission probabilities in system state $\boldsymbol{s} \in \mathcal{S}$ of Case 2 are the second terms in (17), i.e., $\sum_{\theta=1}^{B-n} p_a(k+\theta)p^{\boldsymbol{s}}_c(n+\theta)$ if $k = 0$ and $\sum_{\theta=1}^{B-n} p_a(k+\theta)p^{\boldsymbol{s}}_c(n+\theta)$ if $k \in \mathbb{N}_+$. This completes the proof. ■

Therefore, with the transition probability equation in (17), we can compute the steady-state probability of the queue based on (16). Having the steady-state probabilities of the queue, the PMF of the number of served packets can be derived based on (10), (11), and (12). Consequently, by substituting (10) into (9), $\mathbb{E}\big[\mu^{\boldsymbol{s}}_m\big]$ can be found.

C. No-drop Probability of URLLC Packets

The second performance metric of the URLLC scheduling policy is the URLLC utility, which depends on the probability of not dropping URLLC packets in each mini-slot based on (7).

***Proposition** 2: The probability of not dropping URLLC packets in one mini-slot in system state $\boldsymbol{s} \in \mathcal{S}$ can be expressed as*

$$\mathbb{P}(D^{\boldsymbol{s}}_m = 0) = \sum_{q=0}^{B} p^{\boldsymbol{s}}_Q(q) \sum_{x=q}^{B} p^{\boldsymbol{s}}_c(x), \tag{20}$$

where, $p^{\boldsymbol{s}}_Q(q)$ and $p^{\boldsymbol{s}}_c(x)$ denote the steady state probability of the URLLC queuing backlog and the PMF of provided mini-slots, as introduced in (16) *and* (11)*, respectively. Accordingly, the no-drop probability of URLLC depends on the distribution of URLLC arriving packets and eMBB puncturing weights and thus is the same over different mini-slots of one time-slot (i.e., time-homogeneous over one time-slot).*

Proof: The probability of not dropping URLLC packets can be calculated as

$$\mathbb{P}(D^{\boldsymbol{s}}_m = 0) = \sum_{q\in\mathbb{N}} p^{\boldsymbol{s}}_Q(q)\mathbb{P}(D^{\boldsymbol{s}}_m = 0|Q^{\boldsymbol{s}}_{m-1} = q), \tag{21}$$

where $\mathbb{P}(D^{\boldsymbol{s}}_m = 0|Q^{\boldsymbol{s}}_{m-1} = q)$ denotes the conditional probability that the number of dropped packets is zero given the number of backlogged packets is $Q^{\boldsymbol{s}}_{m-1} = q \in \mathbb{N}$. This probability can be written as

$$\mathbb{P}(D^{\boldsymbol{s}}_m = 0|Q^{\boldsymbol{s}}_{m-1} = q) = \frac{\mathbb{P}(D^{\boldsymbol{s}}_m = 0 \cap Q^{\boldsymbol{s}}_{m-1} = q)}{p^{\boldsymbol{s}}_Q(q)}. \tag{22}$$

If there is no dropped URLLC packet, the number of provided mini-slots for transmitting URLLC packets (i.e., $C^{\boldsymbol{s}}_m$) should be no less than the URLLC queue backlog packets (i.e., $Q^{\boldsymbol{s}}_{m-1}$). Thus,

$$\begin{aligned}&\mathbb{P}(D^{\boldsymbol{s}}_m = 0 \cap Q^{\boldsymbol{s}}_{m-1} = q) = \mathbb{P}(C^{\boldsymbol{s}}_m \geq Q^{\boldsymbol{s}}_{m-1}|Q^{\boldsymbol{s}}_{m-1} = q)\\ &= \sum_{x=q}^{B} \mathbb{P}(C^{\boldsymbol{s}}_m = x), \quad \forall q \in \mathbb{N}, \forall \boldsymbol{s} \in \mathcal{S}, \forall m \in \mathcal{M}. \end{aligned}\tag{23}$$

Based on (21)-(23), the no-drop probability is updated as (20). This completes the proof. ■

IV. Bayesian Optimization for URLLC Scheduling

With the mathematical analysis of the queue dynamics, it becomes apparent that finding the closed-form of the objective function in (6) is infeasible because there is no closed-form solution for the steady-state probabilities of the URLLC queue. In such a case, the optimization problem can be solved as a black-box function where the inputs are the puncturing weights. For each input, the black-box function can be computed. However, the computational cost of acquiring the black-box function for each possible input is high since the representation of the objective function and its derivatives are uncertain and unknown [26], [27]. In other words, during the evaluation, only the outputs of the black-box function for each input will be required and there is no need to analyze the first or second-order derivatives. This is also called "derivative-free" problem, which prevents the use of first and second-order derivatives in optimization algorithms, as required in gradient descent, Newton's, and quasiNewton methods [28]. Thus, it has always been challenging to shorten the evolutionary computation time for stochastic black-box optimization.

Recently, many researchers have been focused on solving the black-box optimization based on the BO algorithm [29]–[31], which is an efficient optimization method to approach a global maximum (or minimum) of an unknown objective function by iteratively developing a global statistical model of it. BO algorithm has two critical elements that need to be chosen appropriately for the different objective functions [31], [32]. The first one is the surrogate model, i.e., a substitute function used to hypothesize the optimized black-box function. Starting with a prior model over the function and a likelihood, at each iteration the surrogate model can compute the posterior distribution by conditioning on the previous evaluations of the objective function. Then, the second critical element of BO, acquisition function, is constructed, which maps the belief of the objective function by measuring the foreground of each position in the input space. After that, the goal is to find the input that maximizes the acquisition function and uses it as a new sampling point for the black-box function evaluation. BO puts the new input in the black-box function to obtain the corresponding output, the surrogate model is then updated. This completes one iteration of the BO loop.

In this section, we first introduce the fundamentals of the surrogate model and acquisition functions. After that, the principles of additive structure to solve high dimensional BO problems will be presented. The proposed solution of (6) using the BO and additive structure algorithms will also be explained.

A. Surrogate Model

We employ the Gaussian process (GP) regression [33] for the surrogate model that provides a Bayesian posterior probability distribution of the objective function for different values of input considering a prior with multivariate Gaussian distribution. GP is a commonly used surrogate model for BO since it can efficiently and effectively summarize a large number of functions, smooth the transition with plenty of observations and thus better quantify the uncertainty in the surrogate model. Besides, the elegant marginalization of the Gaussian distribution contributes to calculating the edges and conditions of the closed-form.

Suppose the black-box objective function in (6) is denoted as

$$f(\boldsymbol{W}) = \sum_{\mathbf{s}\in\mathcal{S}} \mathrm{P}_s(\mathbf{s}) \Big(U^{\boldsymbol{s}}_{\text{embb}} + U^{\boldsymbol{s}}_{\text{urllc}} \Big), \tag{24}$$

where $\boldsymbol{W}$ denotes the matrix of input variables. The evaluation of $f(\boldsymbol{W})$ is restricted to sampling at input $\boldsymbol{W}$ and getting a response. The BS has r groups of input variables $\boldsymbol{W}$ and the corresponding values extracted from $f(\boldsymbol{W})$, denoted as $\mathcal{N}(r) = \big\{ \big(\boldsymbol{W}_1, f(\boldsymbol{W}_1)\big), \ldots, \big(\boldsymbol{W}_r, f(\boldsymbol{W}_r)\big) \big\}$, the resulting prior distribution of (24) with the training set $\mathcal{N}(r)$ under the GP algorithm is assumed as

$$f(\boldsymbol{W}_{1:r}) \sim \mathcal{N}\big(\rho_0(\boldsymbol{W}_{1:r}), K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W}_{1:r})\big), \tag{25}$$

where, $\boldsymbol{W}_{1:r}$ indicates the input sequence $\{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_r\}$, and $f(\boldsymbol{W}_{1:r})$ denotes the corresponding outputs, $f(\boldsymbol{W}_{1:r}) = [f(\boldsymbol{W}_1), ..., f(\boldsymbol{W}_r))]$. Also,

$$\rho_0(\boldsymbol{W}_{1:r}) = [\rho_0(\boldsymbol{W}_1), ..., \rho_0(\boldsymbol{W}_r)] \tag{26}$$

where ρ_0 is the prior mean function, which is set to zero as GPs are flexible enough to model the mean arbitrarily well [34]. Moreover, $K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W}_{1:r})$ is the covariance matrix and represented as

$$K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W}_{1:r}) = \begin{bmatrix} \kappa_0(\boldsymbol{W}_1, \boldsymbol{W}_1) & \ldots & \kappa_0(\boldsymbol{W}_1, \boldsymbol{W}_r) \\ \vdots & \ddots & \vdots \\ \kappa_0(\boldsymbol{W}_r, \boldsymbol{W}_1) & \ldots & \kappa_0(\boldsymbol{W}_r, \boldsymbol{W}_r). \end{bmatrix} \tag{27}$$

In this paper, $\kappa_0(\boldsymbol{W}_{1:r}, \boldsymbol{W}_{1:r})$ follows the *squared exponential* covariance function, which specifies the covariance between each pair of input variables as

$$\kappa_0(\boldsymbol{W}_p, \boldsymbol{W}_q) = \exp\big(-0.5||\boldsymbol{W}_p - \boldsymbol{W}_q||^2\big). \tag{28}$$

BO combines the prior beliefs about $f(\boldsymbol{W})$, updates the priors with the training set, and finally obtains the posterior to approximate the black-box function. Given the observations and Gaussian prior, the posterior distribution of $f(\boldsymbol{W})$ conditioned on $\mathcal{N}(r)$ can be obtained as [28], [34]

$$f(\boldsymbol{W})|\mathcal{N}(r) \sim \mathcal{N}\big(\rho_r(\boldsymbol{W}), \sigma_r^2(\boldsymbol{W})\big) \tag{29}$$

where $\rho_r(\boldsymbol{W})$ and $\sigma_r^2(\boldsymbol{W})$ denote the posterior mean and variance, which are calculated as

$$\rho_r(\boldsymbol{W}) = K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W})^T K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W}_{1:r})^{-1} f(\boldsymbol{W}_{1:r}), \tag{30}$$

and,

$$\begin{aligned} \sigma_r^2(\boldsymbol{W}) &= K_0(\boldsymbol{W}, \boldsymbol{W}) \\ &- K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W})^T K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W}_{1:r})^{-1} K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W}). \end{aligned} \tag{31}$$

Based on the posterior distribution, in BO, an acquisition function will be constructed to quantify the evaluation of the objective function at a new sampling point $\boldsymbol{W}$. In other words, the acquisition function is computed based on the current posterior distribution, then, the next sampling point for observation and evaluation is selected by the maximizer of the acquisition function over the objective domain.

B. Acquisition Function

The acquisition function of BO is used to propose the next sampling point in the search space where there is a trade-off between exploitation and exploration. The goal of both exploitation and exploration is to maximize the acquisition function to determine the next sampling point. However, the former refers to sampling in places where the surrogate model predicts a high objective, and the latter refers to sampling in places where the prediction uncertainty is high.

Given the current observations $\mathcal{N}(r)$, in each iteration the position of the next evaluation point can be decided by maximizing the acquisition function (i.e., denoted as $g(.)$). In particular, the next sample will be selected as

$$\boldsymbol{W}_{r+1} = \arg\max g(\boldsymbol{W}|\mathcal{N}(r)). \tag{32}$$

The standard acquisition functions developed for BO are Expected Improvement (EI), Probability of Improvement (PI), and upper confidence bound (UCB) [28], [31]. The performance of the acquisition function depends on the choice of the surrogate model. The derived expressions of the above three commonly used acquisition functions based on the GP hyperparameters and previous observations set $\mathcal{N}(r)$ are presented as follows.

1) Expected Improvement: The principle of the Expected Improvement algorithm is to maximize the expected improvement over the current best [28], [35], and the expression of Expected Improvement is defined as

$$g_{EI}(\boldsymbol{W}|\mathcal{N}(r)) = \mathbb{E}\Big[\max\Big(f(\boldsymbol{W}) - f_r^*, 0\Big)\Big], \tag{33}$$

where $f_r^* = \max_{j\leq r} f(\boldsymbol{W}_j)$ is the largest observed value in $\mathcal{N}(r)$. Under GP, the expression of (33) is analytically computed based on (29) as

$$g_{EI}(\boldsymbol{W}|\mathcal{N}(r)) = \big(\rho_r(\boldsymbol{W}) - f_r^* - \xi\big)\Phi(Z) + \sigma_r(\boldsymbol{W})\phi(Z), \tag{34}$$

where $Z = \frac{\rho_r(\boldsymbol{W}) - f_r^* - \xi}{\sigma_r(\boldsymbol{W})}$. Moreover, Φ and ϕ are the CDF and Probability Density Function (PDF) of the standard normal distribution, respectively. Parameter $\xi \geq 0$ in (34) determines the amount of exploration during optimization [29], [36]. With higher values of ξ, the acquisition criterion selects the next sampling point which has a larger surrogate variance (i.e., $\sigma_r^2(.)$ in (31)). Conversely, with lower values of ξ, the acquisition criterion results in selecting the next sampling point with a larger surrogate mean value (i.e., $\rho_r(.)$ in (30)).

2) Probability of Improvement: The strategy of Probability of Improvement is to maximize the probability of improving the best current value [37]. Under GP, this algorithm is analytically computed as

$$g_{PI}(\boldsymbol{W}) = \Phi\left(\frac{f_r^* - \rho_r(\boldsymbol{W})}{\sigma_r(\boldsymbol{W})}\right). \tag{35}$$

3) Upper Confidence Bound: The idea of the upper confidence bound is to maximize the regret in the optimization process [28], [38]. Under GP, the upper confidence bound is analytically computed as

$$g_{UCB}(\boldsymbol{W}) = \rho_r(\boldsymbol{W}) - \xi\sigma_r(\boldsymbol{W}). \tag{36}$$

In the simulation section, we implemented all the acquisition functions above to compare the performance among them.

C. Additive Structure

BO is a common technique to optimize an expensive-to-evaluate d-dimensional black-box objective function [32]. However, the algorithm meets challenges when the input space of the objective function is larger since the reliable search and estimation of the surrogate model and acquisition functions in higher dimensional spaces may require more evaluations. In this paper, we apply the additive structure to reduce the high dimensional input space of BO; this technique is an advanced algorithm that can accelerate the search and optimization efficiently [39], [40].

The additive structure assumes that the black-box objective function is an additive function as a sum of sub-functions of all combinations of lower dimensional coordinates. It has been proved in [40] that an additive function only depends linearly on d even though it depends on all d input dimensions. Thus, the working principle of the additive structure in this paper is dividing the high dimensional input space into multiple sub-spaces, so that the number of variables in each sub-space is small, e.g., less than twenty. Accordingly, this technique decomposes the original black-box function in to multiple sub-functions. Since the acquisition function can be optimized component-wise, for each sub-black-box function, we carry out the basic BO optimization, and update the surrogate model and acquisition function in each sub-space. Finally, the posterior distribution of the original black-box function is the sum of the sub-posterior distributions.

Consider the high-dimensional search space of the black-box function $f(\boldsymbol{W})$ is $\mathcal{I} = [0,1]^d$, where $d = B \times N_s$. Based on the additive structure, the original high dimensional input space is decomposed into N_s disjoint sub-spaces, each having merely B dimensions. We assume the decomposed input space is $\mathcal{I} = \{\mathcal{I}_{\boldsymbol{s}_1}, \dots, \mathcal{I}_{\boldsymbol{s}_{N_s}}\}$, where, $\mathcal{I}_{\boldsymbol{s}}$ is the sub-space in system state $\boldsymbol{s}$, $\cup_{\boldsymbol{s}\in\{\boldsymbol{s}_1,\dots,\boldsymbol{s}_{N_s}\}}\mathcal{I}_{\boldsymbol{s}} = \mathcal{I}$, and $\mathcal{I}_{\boldsymbol{s}_i} \cap \mathcal{I}_{\boldsymbol{s}_j} = \emptyset$, $\forall \boldsymbol{s}_i \neq \boldsymbol{s}_j$, $\forall \boldsymbol{s}_i, \boldsymbol{s}_j \in \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_s}\}$. The input variables for the sub-space $\mathcal{I}_{\boldsymbol{s}}$ is the $1 \times B$ vector of the puncturing weights of different channels given $\boldsymbol{s}$, denoted by $\boldsymbol{w}^{\boldsymbol{s}}$.

Accordingly, we decompose $f(\boldsymbol{W})$ into N_s sub-objective functions as the following additive form,

$$f(\boldsymbol{W}) = \sum_{\boldsymbol{s}=\boldsymbol{s}_1}^{\boldsymbol{s}_{N_s}} \mathrm{P}_s(\boldsymbol{s}) f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}}), \tag{37}$$

where $\mathrm{P}_s(\boldsymbol{s})$ is the PMF in system state $\boldsymbol{s}$ and the sub-objective function $f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}})$ is expressed as

$$f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}}) = U^{\boldsymbol{s}}_{\text{embb}} + U^{\boldsymbol{s}}_{\text{urllc}}. \tag{38}$$

At each iteration of BO, the given observations $\mathcal{N}(r)$ is firstly decomposed into N_s groups, $\mathcal{N}^{\boldsymbol{s}}(r)$ denotes the observations set that contains r numbers of input $\boldsymbol{w}^{\boldsymbol{s}}$ and the corresponding output $f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}})$. Based on the decomposed observation groups, the prior distribution of each of the component $f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}}), \forall \boldsymbol{s} \in \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_s}\}$, is obtained independently as

$$f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}}_{1:r}) \sim \mathcal{N}\big(\rho^{\boldsymbol{s}}_0(\boldsymbol{w}^{\boldsymbol{s}}_{1:r}), K^{\boldsymbol{s}}_0(\boldsymbol{w}^{\boldsymbol{s}}_{1:r}, \boldsymbol{w}^{\boldsymbol{s}}_{1:r})\big), \tag{39}$$

Algorithm 1: BO of puncturing weights.

```
Find $f(\boldsymbol{W})$ based on (24) at $r_0$ different eMBB
puncturing weight matrices values. Set $r = r_0$.
while $r \leq R$ do
    Decompose $f(\boldsymbol{W})$ into $N_s$ parts based on (37).
    Decompose both the observations $\mathcal{N}(r)$ and search
    space into $N_s$ parts.
    for $\boldsymbol{s} \in \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_s}\}$ do
        Find the GP prior distribution of $f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}})$.
        Find the posterior mean and variance of $f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}})$
        using $\mathcal{N}^{\boldsymbol{s}}(r)$ and update the posterior distribution.
        Find the next sampling point as
            $\boldsymbol{w}^{\boldsymbol{s}}_{r+1} = \arg\max_{\boldsymbol{w}^{\boldsymbol{s}}} g(\boldsymbol{w}^{\boldsymbol{s}}|\mathcal{N}^{\boldsymbol{s}}(r))$.
        Obtain output sample, i.e., $f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}}_{r+1})$.
    end for
    Update the new sampling point for $f(\boldsymbol{W})$ as
        $\boldsymbol{W}_{r+1} = \cup_{\boldsymbol{s}\in\{\boldsymbol{s}_1,\dots,\boldsymbol{s}_{N_s}\}}\boldsymbol{w}^{\boldsymbol{s}}_{r+1}$.
    Find the output of $\boldsymbol{W}_{r+1}$ by query $f(\boldsymbol{W})$ at $\boldsymbol{W}_{r+1}$.
    Update the observation set as
    $\mathcal{N}(r+1) = \mathcal{N}(r) \cup \{(\boldsymbol{W}_{r+1}, f(\boldsymbol{W}_{r+1}))\}$.
    $r = r + 1$.
end while
```

where $\boldsymbol{w}^{\boldsymbol{s}}_{1:r}$ and $f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}}_{1:r})$ indicate the input sequence and corresponding outputs. The prior mean function is set to zero function, i.e., $\rho^{\boldsymbol{s}}_0(\boldsymbol{w}^{\boldsymbol{s}}_{1:r}) = \boldsymbol{0}$, $\forall \boldsymbol{s} \in \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_s}\}$. Also, $K^{\boldsymbol{s}}_0(\boldsymbol{w}^{\boldsymbol{s}}_{1:r}, \boldsymbol{w}^{\boldsymbol{s}}_{1:r})$ is the covariance matrix, which follows the *squared exponential* covariance function as shown in (28). Then, the mean function and covariance matrix of the original black-box function in (25) is obtained as [40]

$$\rho_0(\boldsymbol{W}_{1:r}) = \sum_{\boldsymbol{s}=\boldsymbol{s}_1}^{\boldsymbol{s}_{N_s}} \rho^{\boldsymbol{s}}_0(\boldsymbol{w}^{\boldsymbol{s}}_{1:r}), \tag{40}$$

$$K_0(\boldsymbol{W}_{1:r}, \boldsymbol{W}_{1:r}) = \sum_{\boldsymbol{s}=\boldsymbol{s}_1}^{\boldsymbol{s}_{N_s}} K^{\boldsymbol{s}}_0(\boldsymbol{W}^{\boldsymbol{s}}_{1:r}, \boldsymbol{w}^{\boldsymbol{s}}_{1:r}). \tag{41}$$

As mentioned before, the original black-box function is decomposed to N_s sub-functions, and then we apply the standard BO to solve each sub-function. We obtain the posterior distribution of $f(\boldsymbol{w}^{\boldsymbol{s}})$ based on the prior distribution and the training set $\mathcal{N}(r)$ as

$$f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}})|\mathcal{N}(r) \sim \mathcal{N}\big(\rho^{\boldsymbol{s}}_r(\boldsymbol{w}^{\boldsymbol{s}}), {\sigma^{\boldsymbol{s}}_r}^2(\boldsymbol{w}^{\boldsymbol{s}})\big). \tag{42}$$

where $\rho^{\boldsymbol{s}}_r(\boldsymbol{w}^{\boldsymbol{s}})$ and ${\sigma^{\boldsymbol{s}}_r}^2(\boldsymbol{w}^{\boldsymbol{s}})$ denote the posterior mean and variance of sub-function $f_{\boldsymbol{s}}(\boldsymbol{w}^{\boldsymbol{s}})$, which are calculated as [40]:

$$\rho^{\boldsymbol{s}}_r(\boldsymbol{w}^{\boldsymbol{s}}) = K^{\boldsymbol{s}}_0(\boldsymbol{w}^{\boldsymbol{s}}, \boldsymbol{w}^{\boldsymbol{s}}_{1:r}) K_0(\boldsymbol{w}_{1:r}, \boldsymbol{w}_{1:r})^{-1} f(\boldsymbol{W}_{1:r}), \tag{43}$$

$$\begin{aligned} \sigma^2_r(\boldsymbol{W}) = {} & K^{\boldsymbol{s}}_0(\boldsymbol{w}^{\boldsymbol{s}}, \boldsymbol{w}^{\boldsymbol{s}}) \\ & - K^{\boldsymbol{s}}_0(\boldsymbol{w}^{\boldsymbol{s}}, \boldsymbol{w}^{\boldsymbol{s}}_{1:r}) K^{\boldsymbol{s}}_0(\boldsymbol{w}_{1:r}, \boldsymbol{r}_{1:r})^{-1} K^{\boldsymbol{s}}_0(\boldsymbol{W}_{1:r}, \boldsymbol{w}^{\boldsymbol{s}}). \end{aligned} \tag{44}$$

The decomposed objective functions are independent, and thus the corresponding surrogate model and acquisition functions are updated in the decomposed sub-space domains with the corresponding posterior mean and variance.

D. Proposed Solution

Here, the proposed solution for (6) using BO and the additive structure is presented. As shown in Algorithm 1, we first create the initial training set $\mathcal{N}(r)$ with r training samples, which includes the input data and the corresponding output data under $f(\boldsymbol{W})$. Each input is randomly generated puncturing weights over all channels and possible system states. We decompose the search space, initial training set, and the original objective function into N_s parts. From step 5 to step 9 in algorithm 1, we find the next sampling point in each sub-space. As step 8 shows, with the posterior mean and variance of each sub-function, we apply the acquisition function in the sub-space to maximally determine where to sample the next sub-objective function. The corresponding output data in each sub-space is found by putting the obtained sampling data in step 8 into the sub-objective function. Then the new sampling data for original objective function is obtained as shown in step 11 and 12. After updating the training set, we iterate between step 2 and step 14 until reaching a maximum number of iterations, i.e, R.

E. Computational Complexity

It has been proved that the computational complexity of the BO algorithm is an order of $O(r^{*3})$, where r^* is the number of samples evaluated until convergence [41], [42]. This computational complexity is dominantly due to the computations required to find the posterior. The computation of the posterior requires the computation of the inverse of the kernel matrix, whose size depends on the number of sample evaluations. More specifically, in the derivation process of $\rho_r(\boldsymbol{W})$ in (30) and $\sigma_r^2(\boldsymbol{W})$ in (31), the computational complexity is dominated by the matrix inverse $K_0^{-1}(\boldsymbol{W}_{1:r}, \boldsymbol{W}_{1:r})$ in (28). Based on [39], [41], [42], the standard methods for matrix inversion of positive definite symmetric matrices require computational time complexity $O(r^3)$ for inversion of an r by r matrix. Therefore, it can be concluded that the complexity of the proposed algorithm only grows cubic in the number of sample evaluations r^*. With the additive structure, the proposed solution in Algorithm 1 further enhances the time efficiency of BO by decomposing the black-box function into several parts and optimizing them in parallel. This will reduce the number of sample evaluations required for convergence.

Besides, the proposed scheme is suitable for time-critical applications. Since this paper considers a stochastic optimization of URLLC scheduling, the algorithm finds an optimal scheduling solution for each possible system state combination. With the optimized results for the system states distribution, the BS could choose the eMBB puncturing weights at the beginning of each decision slot based on which rates the channels undergo. Thus, the complexity of the proposed algorithm does not directly influence the URLLC latency requirement.

V. Results

In this section, we first validate the accuracy of the expressions that analyze the URLLC queue in Section III. The results also show the influence of the queuing mechanism in URLLC requirements. After that, we compare the optimization ability of different acquisition functions with a limited initial training sets. The optimization results of proposed solution are presented in terms of the total normalized eMBB throughput and the utility of URLLC. We evaluated the performance of our proposed queuing mechanism by comparing it with two benchmarks, i.e., the immediate dropping policy in most of the existing works [13]–[18] and the scheduling scheme in [20]. After implementing the joint eMBB-URLLC systems of the benchmarks, the results show that the proposed queuing mechanism in this paper not only enhances the probability of not dropping URLLC packets within the target latency but also improves the total normalized eMBB throughput.

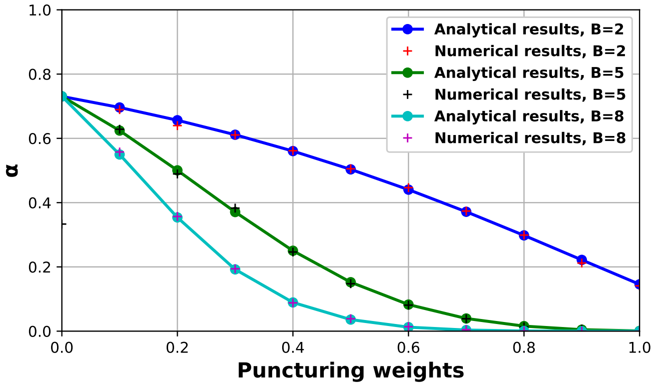

Figure 3. The instantaneous mean latency of served URLLC packets (i.e., α) versus the puncturing weights of all channels (i.e., w).

A. Simulation Setup

According to [15], each URLLC packet should be transmitted within one millisecond, and the scheduling time should be no more than 0.3 milliseconds. 3GPP has suggested that the duration of each eMBB time slot is set to one millisecond with a mini-slot duration of 0.125 milliseconds [11], [15]. Thus, each eMBB slot contains $M = 8$ URLLC mini-slots in this paper. Same as [15], where the authors considered a finite set of eMBB channel rates with equal probability and i.i.d. across users and slots , in this paper, the set of possible channel rates for an eMBB user is assumed as $\mathcal{C} = \{0.1, 0.5, 1\}$ Mbps. We considered all combinations of system states and assumed all possible system states have equal probabilities of being chosen. The URLLC packets were generated randomly in each mini-slot based on the Poisson distribution with the arrival rate λ packets/mini-slot. The maximum number of the iterations in Algorithm 1 is set to $R = 200$. Each experiment was repeated 30 times, and the results show the averaged plots.

B. The Performance of the Queuing Mechanism

First, we investigate the influence of the queue-based scheduling on the URLLC requirements assuming the puncturing weights are the same over all the channels, i.e., $w_b = w, \forall b \in \mathcal{B}$. We define α as the ratio of the expected number of served URLLC packets that were waited for one mini-slot in the queue before transmission (i.e., $\mathbb{E}[Q_{m-1}] - \mathbb{E}[D_m]$) to the expected number of total served URLLC packets as

$$\alpha = \frac{\mathbb{E}[Q_{m-1}] - \mathbb{E}[D_m]}{\mathbb{E}[\mu_m]}. \tag{45}$$

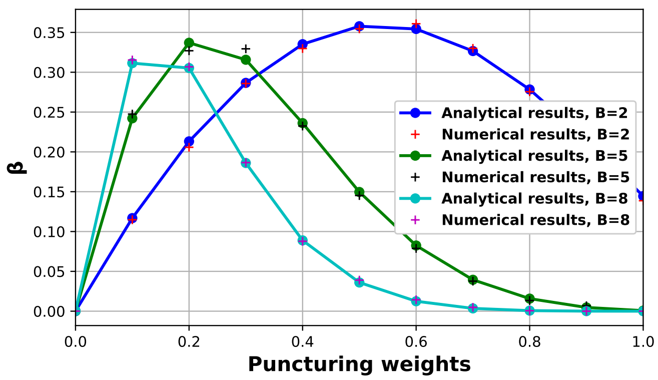

Figure 4. The number of served packets that have waited for two mini-slots (i.e., β) versus the puncturing weights of all channels (i.e., w).

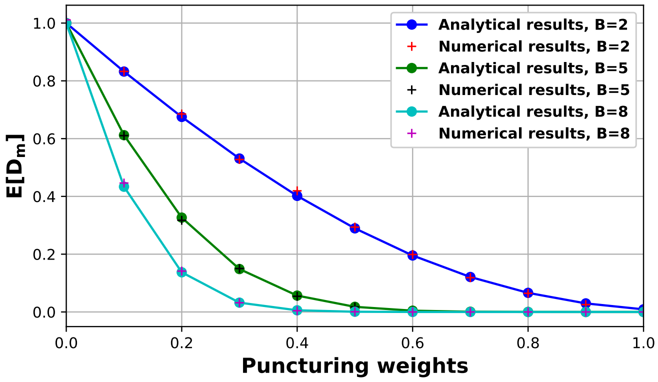

Figure 5. The number of dropped URLLC packets (i.e., $\mathbb{E}[D_m]$) versus the puncturing weights of all channels (i.e., w).

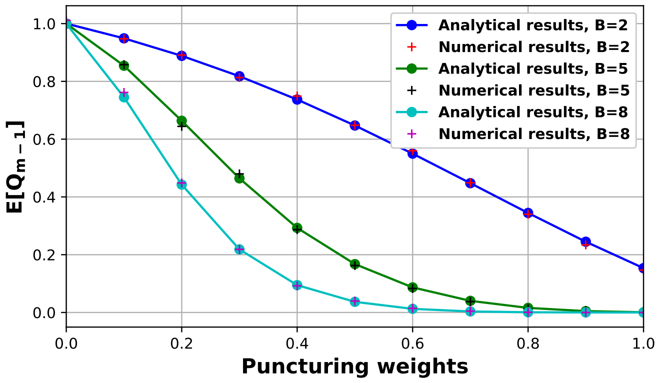

Figure 6. The number of queued URLLC packets (i.e., $\mathbb{E}[Q_{m-1}]$) versus the puncturing weights of all channels (i.e., w).

The measure α shows the percentage of the served URLLC packets that need to be queued to satisfy the ultra-reliability requirement. It also stands for the mean latency of the served packets in each mini-slot. A higher α indicates that with the queue, the BS could transmit more URLLC packets, whereas a small value of α shows the system can transmit URLLC packets directly without queuing.

Fig. 3 shows α versus the puncturing weight (i.e. w) when the URLLC arrival load is $\lambda = 0.5$ packets/mini-slot. This result confirms the validity of the analytical results in Section III, since there is no significant difference between the analytical and numerical results. The mean square errors between analytical and numerical results are below 10^{-3}. This figure also shows that increasing the puncturing weights will reduce the instantaneous mean latency of served URLLC packets to 0.18 mini-slot for $B = 2$ and further to zero if there are more channels available (e.g., $B = 5$ and $B = 8$).

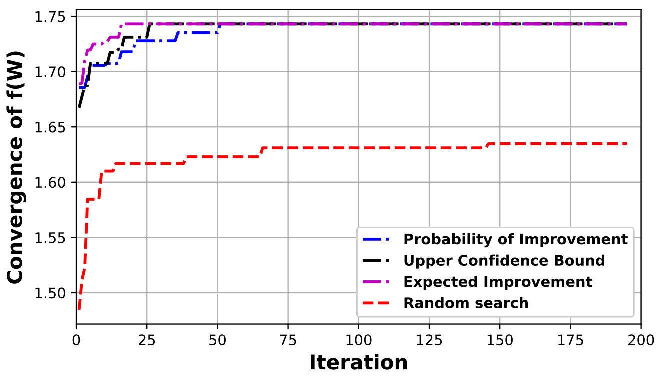

Figure 7. The convergence of the proposed algorithm with different acquisition functions in comparison with the random search when $B = 2$.

To further present how the BS wisely schedules the URLLC packets with the queuing mechanism, Fig. 4 presents the analytical and numerical results of the number of served URLLC packets that have waited one mini-slot in the queue, i.e.,

$$\beta = \mathbb{E}[Q_{m-1}] - \mathbb{E}[D_m]. \tag{46}$$

At first, the values of β increase when increasing the intensity of w. For example, when $B = 2$ and $w = 0.5$, the value of β increases from zero to 0.351 packets. This is because both queued and dropped URLLC packets reduce when the BS decides to puncture more mini-slots. However, the decreasing ratio of $\mathbb{E}[D_m]$ (as shown in Fig. 5) is higher than the decreasing ratio of $\mathbb{E}[Q_{m-1}]$ (as shown in Fig. 6), since with larger values of w, the URLLC packets have a higher chance to be served rather than dropped. However, after the peak points, the BS provides enough mini-slots for URLLC packets, the number of dropped URLLC packets approaches zero (as shown in Fig. 5), and the results show decreasing trends only because of the number of queued URLLC packets reduces.

Furthermore, Fig. 4 also shows that if there are enough channels, the peak points of β are smaller, and the number of queued URLLC packets could quickly approach zero. For example, in Fig. 4, comparing the results of $B = 2$ and $B = 8$, with more transmission channels, the peak point of β is smaller and the trend drops more quickly. Especially when $w = 0.7$, the BS achieves the URLLC requirements with both of the number of queued and dropped packets are zero for $B = 8$. However, for $B = 2$, even with $w = 1$, the URLLC packets still need to be queued to achieve a higher served rate. Besides, the results in Fig. 4, Fig. 5 and Fig. 6 also show that the analytical results match well with the numerical results.

C. Bayesian Optimization Convergence and Performance

In this subsection, we first investigate the convergence of the proposed algorithm to find the optimal solution of the optimization problem (6) as compared to the random search.

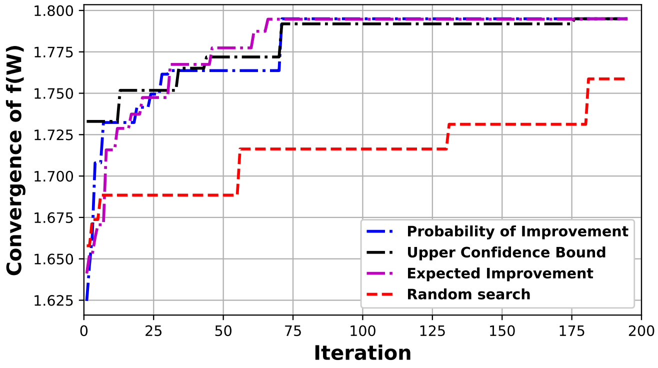

Figure 8. The convergence of the proposed algorithm with different acquisition functions in comparison with the random search when $B = 4$.

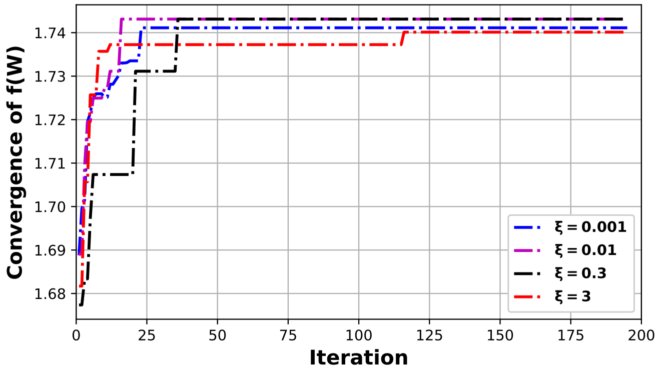

Figure 9. The convergence of proposed algorithm with different trade-off parameter ξ when $B = 2$.

The initial training set $\mathcal{N}(r)$ contains $r = 5$ training samples. The URLLC arrival load is $\lambda = 0.1$ packets/mini-slot. Here, we first set $\xi = 0.01$ as the trade-off parameter between exploitation and exploration since it is the most recommended default value in BO literature [32], [36], [39], [40]. Later, the influence of different values of ξ will be explored in this section. The probability of URLLC successful transmission is $s_{urllc}^{b} = 0.9, \forall b \in \mathcal{B}$.

Fig. 7 and Fig. 8 show convergence behaviors of the proposed algorithm with different acquisition functions in comparison with the random search when transmitting under different numbers of channels. In Fig. 7, for $B = 2$, it is shown that all acquisition functions could sense larger local points than the random search. In this case, the convergence time is short since the original input space is relatively small with $d = 18$. However, by increasing the number of channels $B = 4$ as in Fig. 8, the set of possible system states is significantly expanded ($N_s = 81$). This leads to a higher dimensional input space, i.e., $d = 324$ and hence longer convergence time. Overall, all three acquisition functions could obtain satisfactory optimization results, and the EI acquisition function shows a better convergence performance than others. Thus, we choose the EI acquisition function for the following results.

Here, we also study the impact of the amount of exploration of the EI acquisition function by modifying ξ. Fig. 9 shows the convergence of the proposed algorithm with the EI acquisition function when applying different values of ξ to find the optimal solution of the optimization problem (6). We consider four different levels of trade-off parameters, i.e., $\xi \in \{0.001, 0.01, 0.3, 3\}$. In Fig. 9, for $\xi = 0.01$, the proposed algorithm converges in a few iterations. However, a much lower value of $\xi = 0.001$ puts the focus on 'more local' optimization rather than exploration for global optimization by choosing the next sampling point with higher posterior mean. Therefore, the algorithm does not reach to the optimal solution as for $\xi = 0.01$. With $\xi = 0.3$, the algorithm converges to the optimal point as $\xi = 0.01$ but more slowly compared to the case of $\xi = 0.01$, since increasing the trade-off parameter increases exploration in the acquisition criterion. Increasing ξ to $\xi = 3$ will further slow down the convergence due to more exploration and this results in not reaching the maximum in the first 200 iterations. Therefore, we choose to use $\xi = 0.01$ in the simulations since it will strike a good balance between exploration and exploitation.

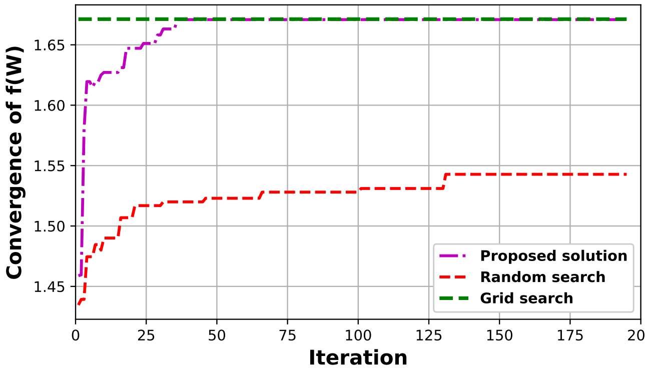

Figure 10. The convergence of proposed algorithm in comparison with the random search and grid search when $B = 1$.

Besides, Fig. 10 shows the performance of the proposed algorithm to achieve the global maximum in comparison with the random search. The global maximum is obtained by evaluating all possible combinations of inputs in the grid search. The number of possible combinations of the puncturing weights in gird search grows exponentially as the number of transmission channels increases. Thus, we consider the case that $B = 1$ in which the number of possible system states is $N_s = 3$ and the dimensional input space is $d = 3$. Let assume the puncturing weights in the gird search belongs to the range of $[0, 1]$ in step size of 0.1. Thus, the number of all possible combinations is $11^3 = 1331$. Fig. 10 confirms the capability of the proposed BO-based algorithm in reaching to the optimal global solution. It is shown that the solution from the Bayesian optimization provides the minimal expected deviation from the global minimum/maximum and the asymptotic density of calculations of the objective function is much greater around the point of global minimum/maximum [43], [44].

In contrast, the random search slowly progress and cannot reach the global point as only parts of the values of the puncturing weights are tried out. The advantage of proposed solution is that it chooses only the relevant search space and discards the ranges that will most likely not deliver the best solution. Thus, instead of randomly choosing the next

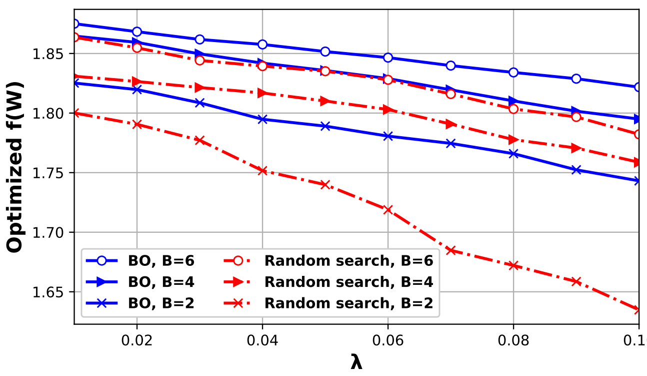

Figure 11. The optimal results versus the URLLC packet arrival rate (i.e., λ) for the proposed queuing scheme when using BO-based proposed solution and random search.

sampling point, the proposed algorithm optimizes the surrogate model and acquisition function according to the previous sampling points at each iteration.

To further investigate the performance of the proposed algorithm, Fig. 11 shows the optimal value of the objective function achieved by the proposed algorithm with EI acquisition function versus the URLLC packet arrival rate. These results verify the effectiveness of the proposed algorithm in comparison with the random search. In particular, the gap in the performance is significant for the higher URLLC arrival rate. This is because the performance is sensitive to the puncturing weights when the BS needs to serve more of URLLC packets.

D. Performance Comparison with Benchmarks

We implemented two benchmarks to highlight the performance of proposed URLLC queuing scheme in the trade-off problem of the joint eMBB-URLLC system.

1) Benchmark 1: Immediate Dropping Scheme: In this scheme, if the newly arrived URLLC packets cannot be transmitted, such packets will be dropped immediately as the losses. Obviously, such policy increases the number of URLLC drops, and to optimize the probability of not dropping URLLC packets, more mini-slots of eMBB services will be interrupted.

2) Benchmark 2: Scheduling Scheme in [20]: In this scheme, each transmission channel has an individual URLLC queue. The newly arrived packets are distributed evenly among these queues, oblivious to the channel gains.

To evaluate the performance of the proposed scheme, in Fig. 12 to Fig. 17, the total normalized eMBB throughput, the URLLC utility, and the URLLC outage probability are analyzed in comparison with two benchmarks. The URLLC arrival rate in Fig. 12 to Fig. 17 changed from 0.01 to 0.1 packets/mini-slot. The objective functions were optimized by the proposed algorithm when using Expected Improvement as the acquisition function. Moreover, we also compare the influence of the probability of URLLC successful transmission on the performance of the proposed scheme. The probability of URLLC successful transmission in Fig. 12 to Fig. 14 is $s^b_{urllc} = 0.9, \forall b \in \mathcal{B}$ and in Fig. 15 to Fig. 17 is $s^b_{urllc} = 0.8, \forall b \in \mathcal{B}$.

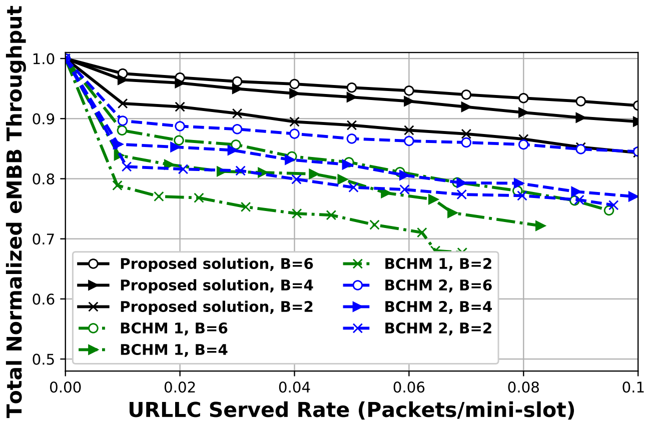

Figure 12. The total normalized eMBB throughput (i.e., U_{eMBB}) versus the URLLC served rate for the proposed solution and benchmarks (BCHM) when $s^b_{urllc} = 0.9, \forall b \in \mathcal{B}$.

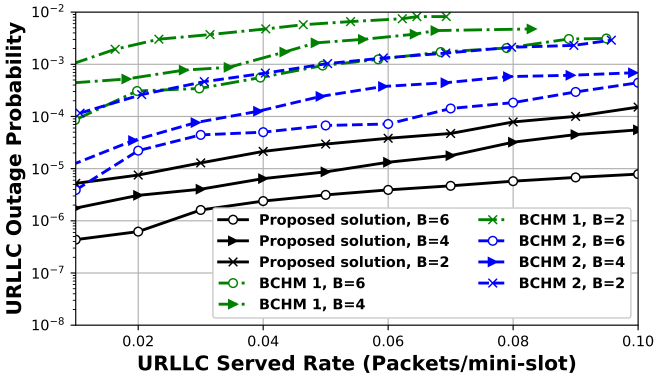

Figure 13. The URLLC outage probability (i.e., P_{out}) versus the URLLC served rates for the proposed solution and benchmarks (BCHM) when $s^b_{urllc} = 0.9, \forall b \in \mathcal{B}$.

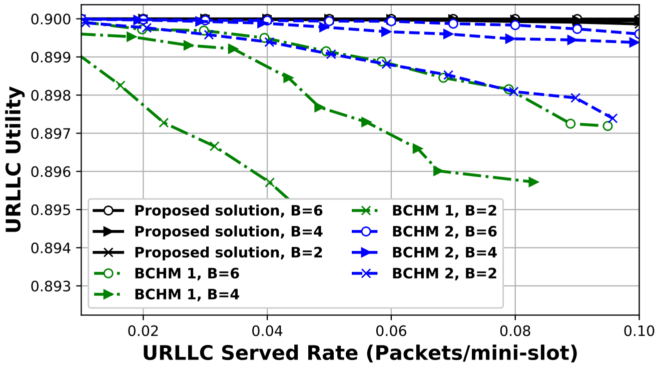

Figure 14. The URLLC utility (i.e., U_{urllc}) versus the URLLC served rates for the proposed solution and benchmarks (BCHM) when $s^b_{urllc} = 0.9, \forall b \in \mathcal{B}$.

Fig. 12 shows the total normalized eMBB throughput (i.e., $U_{\text{eMBB}} = \sum_{\boldsymbol{s} \in \mathcal{S}} P_{\boldsymbol{s}} U^{\boldsymbol{s}}_{\text{eMBB}}$) versus the URLLC served rate (i.e., $\mathbb{E}\left[\mu_m\right]$) under different numbers of channels. The results confirm that the our proposed solution with one queue offers the highest eMBB throughput as compared with Benchmarks 1 and 2. Overall, the proposed solution could maintain the total normalized eMBB throughput above 80%, even when $B = 2$. Although Benchmark 2 performs better than Benchmark 1, the scheduling in [20] is less flexible than our proposed solution

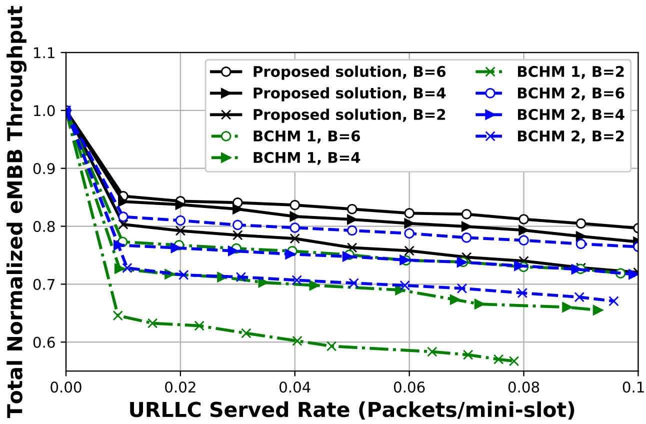

Figure 15. The total normalized eMBB throughput (i.e., U_{eMBB}) versus the URLLC served rate for the proposed solution and benchmarks (BCHM) when $s^b_{urllc} = 0.8, \forall b \in \mathcal{B}$.

with one queue due to even distribution of packets among channels. In the proposed solution with one queue, the URLLC packets are distributed considering the channel conditions of different channels. Besides, as shown in Fig. 12, by queuing the URLLC packets, the total normalized eMBB throughput of both the proposed solution and Benchmark 2 are robust with serving more URLLC packets. In contract, in Benchmark 1, the total normalized eMBB throughput is more sensitive to URLLC served rate.

Furthermore, the performance under all three schemes could be improved by increasing the number of transmission channels. But no-queue scheme is much more sensitive to the number of channels. In Fig. 12, when the URLLC arrival load is $\lambda = 0.02$ packets/mini-slot, the calculated $\mathbb{E}[\mu_m]$ of the no-queue scheme in the cases of $B = 2$ and $B = 6$ are 0.01981 and 0.01631 packets/mini-slot, respectively. And the total normalized eMBB throughput can be improved from 77% to 86% when changing the number of channels form $B = 6$ to $B = 2$. However, the total normalized eMBB throughput of the proposed solution only has a small change.

To further assess the influence on the probability of not dropping URLLC packets, we define the URLLC outage probability as

$$P_{\text{out}} = 1 - \sum_{\boldsymbol{s} \in \mathcal{S}} P_{\boldsymbol{s}}(\boldsymbol{s}) U^{\boldsymbol{s}}_{\text{urllc}}. \tag{47}$$

Fig. 13 shows the URLLC outage probability versus the expected value of served URLLC packets (i.e., $\mathbb{E}[\mu_m]$) under the proposed solution and benchmarks. In Fig. 13, with more channels, all three schemes could reduce the URLLC outage probabilities and enhance the expected values of URLLC served rates. However, the outage probabilities are significantly lower with our proposed solution with one URLLC queue.

Fig. 14 shows the URLLC utility (i.e., $U_{\text{urllc}} = \sum_{\boldsymbol{s} \in \mathcal{S}} P_{\boldsymbol{s}} U^{\boldsymbol{s}}_{\text{urllc}}$) versus the URLLC served rate (i.e., $\mathbb{E}[\mu_m]$) under different numbers of channels. The results confirm that the proposed solution provides the highest URLLC utility compared to the other two benchmarks. Besides, with increasing the URLLC arrivals, the URLLC utility decreases. This is due to the increase of both the URLLC outage probability and the number of failures in transmitted URLLC packets.

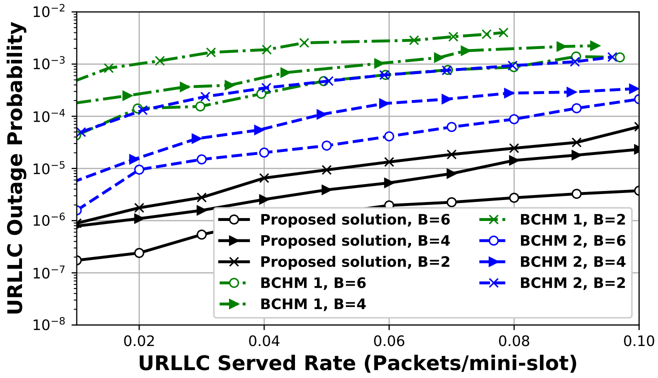

Figure 16. The URLLC outage probability (i.e., P_{out}) versus the URLLC served rates for the proposed solution and benchmarks (BCHM) when $s^b_{urllc} = 0.8, \forall b \in \mathcal{B}$.

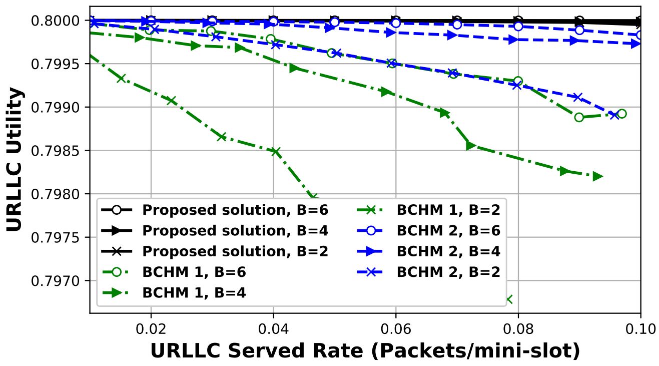

Figure 17. The URLLC utility (i.e., U_{urllc}) versus the URLLC served rates for the proposed solution and benchmarks (BCHM) when $s^b_{urllc} = 0.8, \forall b \in \mathcal{B}$.

Furthermore, we investigate the influence of the probability of URLLC successful transmission by setting $s^b_{urllc} = 0.8, \forall b \in \mathcal{B}$ in Fig. 15-Fig. 17, and compare them to the corresponding results in Fig. 12-Fig. 14 where $s^b_{urllc} = 0.9, \forall b \in \mathcal{B}$. In Fig. 15, the total normalized eMBB throughput values are smaller than the results in Fig. 12. This is because with a lower probability of URLLC successful transmission, the BS will thrive to maintain the URLLC utility by puncturing more eMBB transmissions. Thus, the number of served URLLC packets increases, and Fig.16 shows decreasing trends of URLLC outage probabilities compared to Fig. 13. However, the overall URLLC utility in Fig. 17 is smaller compared to Fig. 14, which implies that the number of successfully transmitted URLLC packets decreases. In contrast, the proposed URLLC queue mechanism robustly provides stable probabilities of not dropping URLLC regardless of S_{urllc}.

Overall, compared to the the benchmarks, the proposed URLLC scheduling policy with the queuing mechanism could enhance the URLLC reliability within the target latency without causing a significant influence on the eMBB transmissions.

VI. Conclusions

In this paper, we developed an optimal URLLC scheduling policy on top of scheduled eMBB channels considering a URLLC queuing mechanism. The scheduling problem has been formulated as a stochastic optimization problem aiming to maximize the total normalized eMBB throughput and URLLC utility. By studying the queue dynamics, we mathematically analyzed the expected value of served URLLC packets and probability of not dropping URLLC packets to derive the objective function. However, it is infeasible to have a closed-form expression due to the computational complexities of deriving the steady-state solution of the URLLC queue. Thus, we applied Bayesian optimization combined with the additive structure algorithm to solve the high dimensional black-box scheduling optimization. The results illustrated that the proposed novel queuing scheme could enhance the probability of not dropping URLLC packets within the target latency requirement in each mini-slot and maintain the total normalized eMBB throughput among all system states. Furthermore, the results also showed that the Bayesian algorithm performed well in finding an optimal point of the black-box function compared with the random search. We also implemented the no-queue scheme and the scheduling scheme in [20] as benchmarks. The results indicated that the optimal URLLC scheduling policy with the proposed queuing scheme has a better flexibility and significantly outperforms the other two scheduling policies in maintaining the total eMBB throughput and reducing the probability of not dropping URLLC packets. According to the limitations of traditional optimization approaches in solving high-dimensional black-box functions, it is necessary to investigate the model-free optimization methods to dynamically and intelligently allocate the limited resources to eMBB and URLLC traffic through interaction with the environment. Besides, the agents of the existing eMBB/URLLC joint scheduling models were unaware of the potential risks caused by the randomness of long-term rewards. Thus, future cellular networks must develop robust joint eMBB/URLLC scheduling strategies given the probability distribution over rewards.

References

[1] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[2] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, "5G URLLC: Design challenges and system concepts," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2018, pp. 1–6.

[3] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega *et al.*, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Communications magazine*, vol. 55, no. 5, pp. 72–79, 2017.

[4] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[5] V. W. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, *Key technologies for 5G wireless systems*. Cambridge university press, 2017.

[6] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[7] H. Chen, R. Abbas, P. Cheng, M. Shirvanimoghaddam, W. Hardjawana, W. Bao, Y. Li, and B. Vucetic, "Ultra-reliable low latency cellular networks: Use cases, challenges and approaches," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 119–125, 2018.

[8] Y. Huang, S. Li, C. Li, Y. T. Hou, and W. Lou, "A deep-reinforcement-learning-based approach to dynamic eMBB/URLLC multiplexing in 5G nr," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6439–6456, 2020.

[9] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 124–130, 2018.

[10] P. Popovski, J. J. Nielsen, C. Stefanovic, E. De Carvalho, E. Strom, K. F. Trillingsgaard, A.-S. Bana, D. M. Kim, R. Kotaba, J. Park *et al.*, "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *Ieee Network*, vol. 32, no. 2, pp. 16–23, 2018.

[11] "Study on new radio access technology; radio interface protocol aspects,version 14.0.0," *3GPP*, p. Rep. TR 38.804, 2017.

[12] K. I. Pedersen, G. Pocovi, J. Steiner, and S. R. Khosravirad, "Punctured scheduling for critical low latency data on a shared channel with mobile broadband," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–6.

[13] M. Alsenwi, N. H. Tran, M. Bennis, A. K. Bairagi, and C. S. Hong, "eMBB-URLLC resource slicing: A risk-sensitive approach," *IEEE Communications Letters*, vol. 23, no. 4, pp. 740–743, 2019.

[14] K. I. Pedersen, G. Pocovi, J. Steiner, and S. R. Khosravirad, "Punctured scheduling for critical low latency data on a shared channel with mobile broadband," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2017, pp. 1–6.

[15] A. Anand, G. de Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 477–490, 2020.

[16] M. Alsenwi, S. R. Pandey, Y. K. Tun, K. T. Kim, and C. S. Hong, "A chance constrained based formulation for dynamic multiplexing of eMBB-URLLC traffics in 5G new radio," in *2019 International Conference on Information Networking (ICOIN)*. IEEE, 2019, pp. 108–113.

[17] J. Zhang, X. Xu, K. Zhang, B. Zhang, X. Tao, and P. Zhang, "Machine learning based flexible transmission time interval scheduling for eMBB and URLLC coexistence scenario," *IEEE Access*, vol. 7, pp. 65 811–65 820, 2019.

[18] J. Li and X. Zhang, "Deep reinforcement learning-based joint scheduling of eMBB and URLLC in 5G networks," *IEEE Wireless Communications Letters*, vol. 9, no. 9, pp. 1543–1546, 2020.

[19] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for embb and urllc coexistence in 5g and beyond: A deep reinforcement learning based approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4585–4600, 2021.

[20] W. Zhang, M. Derakhshani, and S. Lambotharan, "Stochastic optimization of URLLC-eMBB joint scheduling with queuing mechanism," *IEEE Wireless Communications Letters*, vol. 10, no. 4, pp. 844–848, 2021.

[21] G. Americas, "New services applications with 5g ultra-reliable low latency communications," Technical Report, 5G Americas, Tech. Rep., 2018.

[22] P. Popovski, Č. Stefanović, J. J. Nielsen, E. De Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A.-S. Bana, "Wireless access in ultra-reliable low-latency communication (URLLC)," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5783–5801, 2019.

[23] R. Serfozo, *Basics of applied stochastic processes*. Springer Science & Business Media, 2009.

[24] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[25] S. Karlin, *A first course in stochastic processes*. Academic press, 2014.

[26] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de 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.

[27] M. Pelikan et al, "Boa: The Bayesian optimization algorithm," *the genetic and evolutionary computation conference GECCO-99*, vol. 1, pp. 525–532, 2019.

[28] P. I. Frazier, "A tutorial on Bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[29] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.

[30] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[31] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *arXiv preprint arXiv:1206.2944*, 2012.

[32] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams, "Scalable Bayesian optimization

using deep neural networks," in *International conference on machine learning*. PMLR, 2015, pp. 2171–2180.

[33] M. Deisenroth and J. W. Ng, "Distributed Gaussian processes," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1481–1490.

[34] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[35] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of bayesian methods for seeking the extremum," *Towards global optimization*, vol. 2, no. 117-129, p. 2, 1978.

[36] D. J. Lizotte, "Practical bayesian optimization," 2008.

[37] H. J. Kushner, "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise," 1964.

[38] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[39] Z. Wang, C. Gehring, P. Kohli, and S. Jegelka, "Batched large-scale Bayesian optimization in high-dimensional spaces," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 745–754.

[40] K. Kandasamy, J. Schneider, and B. Póczos, "High dimensional Bayesian optimisation and bandits via additive models," in *International conference on machine learning*. PMLR, 2015, pp. 295–304.

[41] G. Lan, J. M. Tomczak, D. M. Roijers, and A. Eiben, "Time efficiency in optimization with a bayesian-evolutionary algorithm," *Swarm and Evolutionary Computation*, vol. 69, p. 100970, 2022.

[42] M. Malu, G. Dasarathy, and A. Spanias, "Bayesian optimization in high-dimensional spaces: A brief survey," in *2021 12th International Conference on Information, Intelligence, Systems Applications (IISA)*, 2021, pp. 1–8.

[43] J. Mockus, "The bayesian approach to global optimization," in *System Modeling and Optimization*. Springer, 1982, pp. 473–481.

[44] J. B. Mockus and L. J. Mockus, "Bayesian approach to global optimization and application to multiobjective and constrained problems," *Journal of optimization theory and applications*, vol. 70, no. 1, pp. 157–172, 1991.