ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Computation Offloading in Heterogeneous Vehicular Edge Networks: On-line and Off-policy Bandit Solutions

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Bozorgchenani, A., Maghsudi, S., Tarchi, D., Hossain, E. (2022). Computation Offloading in Heterogeneous Vehicular Edge Networks: On-line and Off-policy Bandit Solutions. IEEE TRANSACTIONS ON MOBILE COMPUTING, 21(12), 4233-4248 [10.1109/TMC.2021.3082927].

Availability:
This version is available at: https://hdl.handle.net/11585/820940 since: 2022-11-23

Published:
DOI: http://doi.org/10.1109/TMC.2021.3082927

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

12 October 2024

This is the final peer-reviewed accepted manuscript of:

A. Bozorgchenani, S. Maghsudi, D. Tarchi and E. Hossain, "Computation Offloading in Heterogeneous Vehicular Edge Networks: On-Line and Off-Policy Bandit Solutions," in *IEEE Transactions on Mobile Computing*, vol. 21, no. 12, pp. 4233-4248, 1 Dec. 2022.

The final published version is available online at:

https://doi.org/10.1109/TMC.2021.3082927

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

Computation Offloading in Heterogeneous Vehicular Edge Networks: On-line and Off-policy Bandit Solutions

Arash Bozorgchenani, *Member, IEEE,* Setareh Maghsudi,
Daniele Tarchi, *Senior Member, IEEE,* and Ekram Hossain, *Fellow, IEEE*

Abstract—With the rapid advancement of Intelligent Transportation Systems (ITS) and vehicular communications, Vehicular Edge Computing (VEC) is emerging as a promising technology to support low-latency ITS applications and services. In this paper, we consider the computation offloading problem from mobile vehicles/users in a heterogeneous VEC scenario, and focus on the network- and base station selection problems, where different networks have different traffic loads. In a fast-varying vehicular environment, computation offloading experience of users is strongly affected by the latency due to the congestion at the edge computing servers co-located with the base stations. However, as a result of the non-stationary property of such an environment and also information shortage, predicting this congestion is an involved task. To address this challenge, we propose an on-line learning algorithm and an off-policy learning algorithm based on multi-armed bandit theory. To dynamically select the least congested network in a piece-wise stationary environment, these algorithms predict the latency that the offloaded tasks experience using the offloading history. In addition, to minimize the task loss due to the mobility of the vehicles, we develop a method for base station selection. Moreover, we propose a relaying mechanism for the selected network, which operates based on the sojourn time of the vehicles. Through intensive numerical analysis, we demonstrate that the proposed learning-based solutions adapt to the traffic changes of the network by selecting the least congested network, thereby reducing the latency of offloaded tasks. Moreover, we demonstrate that the proposed joint base station selection and the relaying mechanism minimize the task loss in a vehicular environment.

Index Terms—Vehicular Edge Computing (VEC), computation offloading, heterogeneous networks, off-policy learning, on-line learning, bandit theory.

1 Introduction

During the past few years, edge computing has emerged as a distributed computing paradigm that brings the capabilities and resources of the cloud towards the network edge [1]. Mobile Edge Computing (MEC) or, as renamed by ETSI, multi-access edge computing, offers an ultra-low latency environment with high bandwidth and real-time access to network resources in a mobile network [2], [3].

- *A. Bozorgchenani was with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy. He is now with the Department of Computing and Communications, Lancaster University, the UK.*
- *D. Tarchi is with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy.*
- *S. Maghsudi is with the Department of Computer Science, University of Tübingen, Germany.*
- *E. Hossain is with the Department of Electrical and Computer Engineering, University of Manitoba, Canada.*

E. Hossain's work was supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).
A major part of the work was done during A. Bozorgchenani's visit to the University of Manitoba, Canada, supported by a Marco Polo Scholarship funded by the University of Bologna, Italy, and by the project "GAUChO - A Green Adaptive Fog Computing and Networking Architecture" funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W_004. The work of S. Maghsudi was supported by Grant 01IS20051 from the German Federal Ministry of Education and Research (BMBF).

Vehicles have been evolving since the second industrial revolution and their role in modern life is imperative. With the rapid technological advancements in Intelligent Transportation Systems (ITS) technologies, vehicles are equipped with wireless communication capabilities for both intra-vehicle and inter-vehicle communications. ITS technologies support a plethora of applications including those for road safety, smart transportation, and location-dependent services [4]. Vehicular Edge Computing (VEC) has been widely discussed in the literature [5], [6], where the computing infrastructures at both the network and the vehicles are used by mobile users. In essence, VEC combines the concepts of vehicular networking and MEC. If performed suitably, task offloading reduces the energy consumption and speeds up the response time of applications in a VEC scenario [7], [8], [9].

In vehicular networks, Road Side Units, referred to as Base Stations (BSs) throughout this paper, provide reliable wireless access in their coverage ranges. The edge computing servers for the mobile vehicles are supposed to be co-located with the BSs. We study a task offloading problem from vehicles to the edge computing servers, where the tasks generators can be the driving systems or the passengers of the vehicles. We consider a multi-operator heterogeneous scenario consisting of several access networks with different characteristics and several BSs with different coverage areas. Due to the mobility of the vehicles/users, the traffic load across the networks is time-varying. In such

a dynamic environment, vehicles select the best network to perform computation offloading. This might yield intensive traffic load and congestion at the edge server-side, thereby harming the offloading performance. This paper proposes an offloading decision framework that can dynamically select the most suitable network.

Task offloading in VEC involves several challenges. For example, a low latency, which is crucial to ensure an acceptable Quality of Experience (QoE), is difficult to achieve. In a computation offloading scenario, the latency consists of different components including the communication latency, the processing latency, and the waiting time at the computing buffers. In MEC, the waiting times at the edge computing servers often dominate the transmission times [10]; therefore, we concentrate on the analysis of waiting time for an offloaded task, referred to as the *latency of computation offloading*.

For a reliable offloading decision, the vehicle should be aware of both task and network parameters. While task size and number of operations for processing can be apriori determined by the vehicle, the waiting time at the edge server-side, depending on the network load, is unknown. This lack of knowledge weakens the decision on task offloading.

The VEC scenario is characterized by frequent changes in network availability and traffic, mainly due to vehicle mobility and the random characteristics of the wireless medium. In this work, we focus on a non-stationary, or more specifically, a *piece-wise stationary* scenario, where the traffic load of the networks remains constant within some period, and changes at some unknown time instant called *change points*. The *piece-wise stationary* scenario is a suitable model to formalize the non-stationarity of the vehicular environment in terms of network congestion and data traffic. In such a scenario, detecting change points, which potentially alter the offloading decisions, is challenging. To develop efficient decision-making policies, the availability of a data set, including the previous offloading decisions, the network's characteristics, and the traffic's statistics for each network, can be quite useful. Machine learning is a key method for extracting and learning useful information from data to develop decision-making policies [11]. Motivated by the above-mentioned factors, i.e., the time-variant statistical characteristics of the vehicular environment, lack of perfect knowledge before decision-making, and the inevitability of exploration-exploitation trade-off in such a dynamic environment, we propose two machine learning-based methods for making decision on offloading. Among several machine learning methods, we focus on *multi-armed bandit (MAB)* theory, as it enables the user/vehicle to learn through time by exploiting the bandit feedback. In bandit theory, the decision-maker observes some context, performs an action, and receives some feedback in terms of cost or utility. The decision-maker then uses the history of action-outcome pairs to improve its future performance.

The policies used in such an interactive environment can be deployed either on-line or off-line manners, each having its pros and cons [12]. In particular, the on-line approach is suitable for scenarios where no prior information is available to the decision-maker. Such an approach, however, might require excessive time for convergence to the optimal decision. To implement the off-policy method, the existence of the data set is necessary. Such a data set shall improve the performance by enhancing the decision-making policy. The improvement depends on several factors such as the quality of the data, the performance of the underlying decision-making policy, the problem setting such as the dynamics of the environment, and the like.

In this paper, we propose an on-line and and an off-policy algorithm each with two phases: decision making on where to offload, and task offloading. In both the algorithms, the vehicle (or the vehicular user) takes advantage of the historical offloading records. While in the on-line algorithm, the vehicle adapts its decisions over time, in the off-policy approach, the decision is made once for a given amount of time. Using the proposed solutions, a vehicle is able to perform an efficient network selection for computation offloading in presence of unpredictable or non-stationary traffic load at the edge computing servers. The main contributions of this paper can be summarized as follows:

- We model the computation offloading problem in a VEC environment by considering a *piece-wise stationary scenario*, which is a good approximation of the dynamicity in a vehicular network. To the best of our knowledge, no previous work in the literature has investigated the task offloading problem in a piece-wise stationary VEC scenario.
- We develop an on-line network selection scheme based on congestion and traffic patterns in a multi-access edge computing network using the MAB theory, which is a suitable mathematical framework for problems with no prior information and limited feedback. The proposed solution aims at minimizing the latency for the offloaded tasks.
- We propose an off-policy network selection approach by exploiting the historical offloading data set. The proposed approach first detects the change points of the piece-wise stationary scenario. Then, the network selection is performed based on the developed off-policy approach aiming at minimizing the latency for the offloaded tasks. To the best of our knowledge, this is the first work to study off-policy learning for task offloading in a VEC environment.
- We propose a BS selection method based on the sojourn time of the vehicle in the selected network. In addition, we propose a relaying mechanism to minimize the probability of task loss during the offloading procedure due to mobility of the vehicles. The relaying mechanism regards the task size and the application types as random variables that provides generality to the solution.
- We perform extensive numerical analysis to demonstrate the effectiveness of the proposed approaches to adapt to the changes in traffic load in the considered piece-wise stationary scenario. Adapting to the environment reduces latency while guaranteeing a tolerable task loss.

The rest of the paper is organized as follows. In Section 2, we review the state of the art. In Section 3, we describe the system model. Section 4 introduces our proposed on-line and off-policy network selection approaches. In Section 5,

we introduce the BS selection and relaying methods. In Section 6, we present the performance evaluation results. Section 7, concludes the paper.

2 Related Works

Computation offloading in a mobile environment has been investigated intensively in recent literature. In [7], the authors study the problem of energy conservation on mobile devices by offloading tasks to the infrastructure-based cloud. Reference [8] proposes a heuristic offloading decision algorithm that optimizes the offloading decision while allocating communication and computation resources. A similar work is [13], where the authors address the joint optimization of the offloading decisions, the allocation of computation resources, transmit power, and radio bandwidth. The problem of joint optimization of offloading decision, resource allocation, and content caching strategy is considered in [9]. In [14], the authors target a joint optimization of offloading decision making, computation resource allocation, resource block assignment, and power distribution in a mixed fog and cloud network. In [15], the authors propose a partial offloading solution considering the sojourn time of the mobile devices in order to minimize the latency and task loss. Reference [16] couples network selection with service placement. The authors propose an online heuristic, where a centralized operator at each time instant chooses a base station and a proper service placement for each user. In [17], the authors consider a predictive Lyapunov optimization technique and propose a greedy predictive energy-aware network selection and resource allocation scheme. They model a mobile environment where the operator can adjust between power consumption and traffic delay.

In [18], the authors consider a dynamic environment with different wireless networks among which the decision is to select the network reducing the execution cost. Reference [19] proposes a fog offloading scheme in a mobile environment. In brief, if the sojourn time of a mobile user is less than the transmission time, it performs a local computation. In case of offloading, if the computation time is less than the sojourn time, the base station sends the result back to the user, otherwise, the task will be migrated to the cloud for relaying the result to the destination BS of the user. The objective of the work is to minimize cloud migration by proposing a generic-based solution. However, no queuing model is considered in these works and the task waiting time for processing is ignored.

Furthermore, there exists a rich body of the research work that studies the task offloading problem in Vehicular Adhoc Networks (VANETs) and VEC. In [20], the authors propose a MEC-based computation offloading framework to minimize the vehicles' cost for task offloading while guaranteeing processing delay. In [21], to maximize the economical profit of service providers while maintaining low delays, the authors develop a game-theoretical approach that jointly optimizes the task offloading decision and computation resource allocation. Reference [22] proposes to utilize vehicles as the infrastructure for communication and computation. The authors then analyze both scenarios of moving and parked cars as infrastructure. In [23], the authors formulate a dual-side cost minimization, which jointly optimizes the offloading decision and local CPU frequency on the vehicles' side and the radio resource allocation on the servers' side. A graph-based scheduling scheme for V2V and V2I communication has also been studied in [24]. Most of the research works described before neglect the waiting time and the reception time when developing models and solutions for network- or BS selection. Moreover, often they ignore the fact that the selected BS might not be able to process the offloaded task within the sojourn time of the vehicle.

Recently, learning theory has been applied by several authors to address the computation offloading problem. In [25], the authors target the minimization of the delay and the utilization of physical machines for task offloading in mobile cloud computing. They propose a two-layer Reinforcement Learning (RL) structure, where a deep RL method is used to select the optimal cluster in the first layer. In the second layer, a Q-learning approach is used to select the optimal physical machine in the selected cluster. However, the offloading is to the cloud and there is no mobility considered in the scenario. Similarly, in [26], the authors aim at minimizing the latency for task assignment to the servers and propose an RL method. However, this work lacks a detailed formulation of the delay model and considers a simplified scenario. In [27], the authors have formulated user to server allocation as an auction problem and proposed two allocation and payment deep neural networks to maximize the profit of edge servers and satisfy the energy and delay constraints of users.

Reference [28] utilizes the multi-arm bandit (MAB) theory to minimize delay in computation offloading. The proposed MAB solution is for a Vehicle-to-Vehicle (V2V) scenario and assumes that all offloaded tasks are received from the same vehicle when offloaded; however, due to the (fast) mobility, the vehicles cannot receive the result while they are within the sojourn time of the processing vehicle. Moreover, the system model does not include any queuing model, although every processing vehicle might receive multiple tasks. Besides, the environment is stationary, which is not realistic in a fast-varying vehicular environment. A similar work is [29], where the authors consider network load as a parameter in the Upper Confidence Bound (UCB) function such that offloading to new nodes is favorable when the network load is low. By this, when the network load level is low, the exploration is emphasized. The exploitation has a higher weight in heavy load scenarios. Reference [30] studies the task assignment problem in small cell networks. The authors calculate the probability of assigning the tasks to each network based on bandit feedback. They also propose a greedy collaborative task offloading scheme among small cell networks considering the selection probability. In [31], the authors formulate an online optimization problem for task offloading in MEC. They target the queue stability and delay minimization by controlling the portion of data to be analyzed locally and to be offloaded. However, in both of the aforementioned papers, the methods and also the considered scenarios are different from ours.

References [32], [33], and [34] apply piece-wise stationary bandit models, respectively, for server selection, small cell planning, and channel selection for power line communication. The problems under investigation, are, however, not related to the one considered in this paper. Task alloca-

tion under uncertainty is also considered in [35]; however, the approach is one-shot based on the expected utility, rather than being iterative.

To summarize, several state-of-the-art research assume the availability of knowledge about offloading and network parameters, although such an assumption is not realistic in fast-varying heterogeneous environments. Moreover, most of the existing learning-based offloading solutions do not have suitable policies for decision making in a dynamic vehicular environment. A large body of the current literature concerning distributed computation offloading in vehicular scenarios ignore the non-stationarity of the medium. That includes, for example, the dynamic variation of traffic load at the edge servers (e.g. due to vehicles' mobility). In this work, we take these non-stationarity and traffic dynamicity into account to develop two on-line and off-policy learning-based solutions.

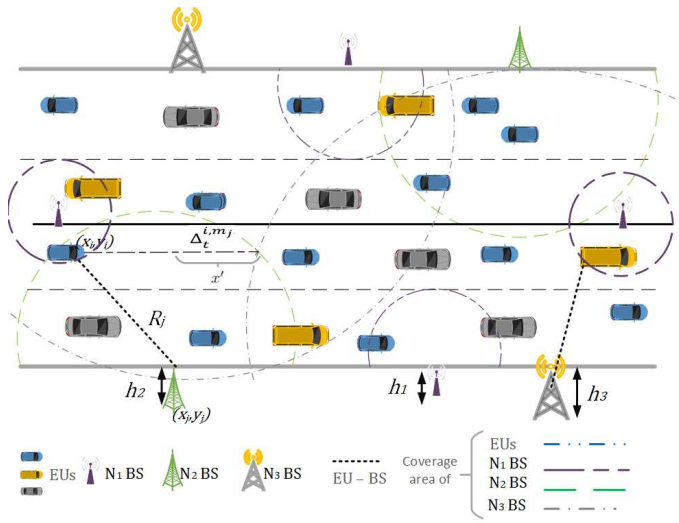


Fig. 1. Computation offloading in a vehicular edge computing scenario.

3 System Model

In this section, we first describe the network model and the offloading environment. Afterwards, we will explain the task computation, communication, and queuing models.

3.1 VEC Model

We consider a two-layer architecture for VEC, where there are several Edge Units (EUs) belonging to the set of EUs $\mathcal{U}$, in the first layer and u_i is the EU of interest. All the EUs have certain computational and storage capabilities. The second layer includes M wireless networks, each having certain number of BSs [18]. $\Im_m = \{N_1, \ldots, N_M\}$ shows the set of all networks, where N_m represents the m-th network. Moreover, $\Im_j^m = \{N_{m_1}, \ldots, N_{m_J}\}$, shows the set of all BSs of the m-th network type, where N_{m_j} identifies the jth BS of the mth network. Every EU can communicate with a BS by means of cellular-based communications. Each BS is equipped with a number of computational servers and therefore can process different tasks. In general, the BSs have higher computational capabilities compared to the EUs. They can aggregate the EUs' traffic requests and process the offloaded tasks. The BSs of each network *type* are homogeneous, meaning that they have the same computational capability, edge server traffic model, and coverage. In contrast, the BS characteristics for different network types vary from one network to another. Note that in a certain coverage area, BSs of different network types (e.g., owned by the same network operator) and also BSs of the same type (e.g., owned by different network operators) exist.

We consider an urban scenario as shown in Fig. 1, where the vehicles act as EUs. Each vehicle i moves with some velocity v_i. The vehicles move either from left to right or in the opposite direction, depending on their location. The speed and the direction of each vehicle remain fixed through time [20], [36]. A *task loss* occurs if an offloading vehicle cannot receive the result of the offloaded task due to its mobility, i.e., due to the short sojourn time. In a heterogeneous wireless access network, the BSs of different networks overlap while covering the entire area. In addition to different coverage and computational capacities, heterogeneous networks can also face different traffic demands due to the offloaded tasks and therefore different levels of congestion at the edge servers. We also assume a full task offloading scenario.

Based on the discussion above, the offloading decision in dynamic vehicular environments is twofold. First, an EU selects the optimal network to offload its task, based on the level of congestion. Afterwards, it selects a BS in the selected network type having a high probability on the reception of the result of the offloaded task. Since in VEC scenarios, latency is mainly dominated by waiting time, in the rest of the paper we focus on waiting time minimization. *Thus, the goal is to address waiting time minimization problem, by a suitable selection of the network and BS for computation offloading.*

3.2 Task Computation and Communication Model

The computation time for the lth task generated by the EU is defined as:

$$T_l^{\text{com}} = \begin{cases} \frac{O \cdot \rho_l^{\text{up}}}{\eta_i} & \text{local computation} \\ \frac{O \cdot \rho_l^{\text{up}}}{\eta_{m_j}} & \text{otherwise,} \end{cases} \tag{1}$$

where O represents the number of operations required for computing one byte, ρ_l^{up} is the task size of the lth task generated by the EU in byte, and η_* is the Floating-point Operation Per Second (FLOPS) depending on the CPU of the device, which could be either a local processor or a processor at the m_jth BS.

The transmission time of the lth task from the EU to the m_jth BS is given by

$$T_{l,t}^{\text{trans},m_j} = \frac{\rho_l^{\text{up}}}{r_{m_j,t}^{\text{up}}}, \tag{2}$$

where $r_{m_j,t}^{\text{up}}$ is the up-link data rate of the link between the EU and the m_jth BS at time instant t. The result of the processed task should be sent back from the m_jth BS to the EU, leading to a reception time as

$$T_{l,t}^{\text{rec},m_j} = \frac{\rho_l^{\text{dl}}}{r_{m_j,t}^{\text{dl}}}, \tag{3}$$

where $r^{\text{dl}}_{m_j,t}$ is the down-link data rate and ρ^{dl}_l is the size of the result of processing.

We assume that there is no inter-cell interference in the up-link transmissions (e.g. due to orthogonal channel sets used by different BSs), whereas the EUs may experience intra-cell interference. We make the following assumption.

Assumption 1. *The EU u_i inside a cellular network experiences independent and identically distributed interference signals caused by other EUs. However, inter-cell interference can be neglected due to inter-cell coordination.*

In a dense vehicular network, a fully-orthogonal transmission is often not possible due to mobility and density; consequently, we consider intra-cell interference (as in [37]). It is worth noting that our approach is generic and applicable also in combination with other interference models. Focusing on a specific time instant when EU i is inside the m_jth cell, we define $\mathbb{I}^{i,\iota} = 1$, where $i, \iota \in \mathcal{U}, i \neq \iota$, if the ith EU and ιth EU cause interference to each other. Thus the interference received by the ith EU yields

$$I^i_{m_j,t} = \sum_{\iota} \mathbb{I}^{i,\iota} \beta^{\iota,m_j}_t P^{\iota}_{\text{trans}}, \tag{4}$$

where β^{ι,m_j}_t is the path-loss attenuation factor. It is a function of the distance between the BS and the vehicle, shown as d^{i,m_j}_t. Moreover, P^{ι}_{trans} is the transmission power of the interferer. The distance between the ith EU and the m_jth BS is given by

$$d^{i,m_j}_t = \sqrt{h^2_{m_j} + |y_{m_j} - y_{i,t}|^2 + |x_{m_j} - x_{i,t}|^2}, \tag{5}$$

where $\{x_{i,t}, y_{i,t}\}$ and $\{x_{m_j}, y_{m_j}\}$ are the positions of the ith EU at time t and the m_jth BS, respectively. Moreover, h_{m_j} is the height of the m_jth BS.

If B_m represents the bandwidth of a channel, the transmission rate in the up-link from the ith EU to the m_jth BS at time instant t can be given as

$$r^{\text{up}}_{m_j,t} = B_m \log_2 \left(1 + \frac{\beta^{\iota,m_j}_t P^i_{\text{trans}}}{I^i_{m_j,t} + \zeta}\right), \tag{6}$$

where P^i_{trans} is the transmission power of the ith EU, ζ is the noise power defined as $\zeta = N_0 B_m$, where N_0 is the noise power spectral density. When sending the processed tasks back to the vehicles, the transmission rate in the down-link, i.e., from the m_jth BS to the ith EU, is calculated similarly using the BS transmission power except that there will be no interference due to the orthogonal channels used by different BSs. We assume that the channel is quasi-static during the transmission- and reception periods.

3.3 Traffic Model at Edge Computing Servers

We consider a Poisson-Exponential (PE) queuing model [38], as described in the following. Servers have a buffer large enough to queue all tasks. The arrival of tasks to the BSs that belong to the mth network follows a Poisson distribution $\text{Pois}(\lambda_m)$. Moreover, the service time for each task follows an exponential distribution $\text{Exp}(\mu_m)$. Let $\vartheta^m_t(\lambda_m, \mu_m)$ denote the amount of tasks in the queue of the mth network at time instant t. $T^l_{w_m}(\vartheta_t)$[1] shows the total waiting time of the lth task offloaded to m-th network. The BSs belonging to the same network type are assumed to be statistically identical, meaning that they have the same task-arrival and departure parameters.

Computational offloading involves two phases: (i) Network selection based on congestion characteristics; and (ii) BS selection in the selected network based on the sojourn time. In the following, we describe our proposed network and BS selection procedures.

4 Network Selection

Since we assume that the BSs of each network have the same statistical parameters, they suffer the same congestion level on average, and therefore, the first step of the offloading decision boils down to network selection. In this section, we formulate the network selection problem with the aim of minimizing the task waiting time at the edge servers. We then propose two learning approaches. In the first approach, an on-line UCB algorithm is developed, while in the second approach, an off-policy method is developed to select the best network by incorporating a change point detection mechanism. For a better clarity, in the following, we list the key notations of this section in Table 1.

4.1 Problem Formulation for Network Selection

We consider a time-slotted system, where time slot t starts at time instant t. We assume that tasks are always generated at the beginning of a time slot. Then each time slot becomes a decision round. Hence, we denote both time slot and decision round as t. Let $\mathbb{Q}^m_t$ be an indicator function that returns 1 if EU offloads to the mth network at the decision round of t. During T rounds of offloading decision, the EU's objective is to offload a task l to a network such that its expected waiting time is minimized. Therefore, the problem can be stated as follows:

$$\mathbf{P1}: \underset{m \in \Im_m}{\text{minimize}} \left\{ \sum_{t=1}^{T} T^l_{w_m}(\vartheta^m_t) \right\} \tag{7}$$

subject to

$$\mathbf{C1}: \sum_{m \in \Im_m} \mathbb{Q}^m_t = 1, \quad \forall t \tag{8}$$

Constraint (8) guarantees that each EU offloads to only one network. The actions of other users/agents are abstracted in the queuing process; hence, ϑ^m_t models the traffic generated by other EUs in each network.

The waiting time for an offloaded task depends on the load of the selected network. In case of the availability of network load information, the EU selects the network as $\text{argmin}_m\{T^l_{w_m}(\vartheta^m_t)\}$. However, the EU is not aware of the offloading decisions of other EUs and therefore the BSs' queue status. Therefore, in the following, we develop a learning-based solution for network selection for computation offloading in a VEC environment.

1. More specifically, a task is offloaded to a BS, m_j, in the network m; however, for the sake of simplicity of the notation we omit the BS index.

TABLE 1
Nomenclature (in the order of appearance)

Notation	Description
$\mathbb{Q}_t^m$	Indicator function for offloading to network m
$T_{w_m}^l$	Waiting time for task l in network m
c_t^m	Instantaneous cost function
λ_m	Mean arrival rate of the tasks in network m
μ_m	Mean service time of the tasks in network m
τ	UCB window length
$C_t^m(\tau)$	Number of selection of network m in τ
C_t	Number of offloaded tasks by round t
$\hat{c}_t^m(\tau)$	SW-UCB cost index
$\bar{c}_t^m(\tau)$	Average accumulated cost
ξ	A constant weight on SW-UCB index
β	Upper bound on exploration factor on SW-UCB index
$L_t^*(m)$	Expected cost of offloading to the optimal network
$L_t(m)$	Expected cost of offloading to the selected network
R_T^{On}	Cumulative regret of on-line approach
$\mathcal{B}$	Set of change points
b_υ	υ-th change point
$\mathcal{Z}$	Set of intervals
z_ψ	ψ-th interval
ω	Length of an interval
ν^ψ	Vector context for each interval in the logs
χ	Context distribution
D^ψ	Cost distribution in each interval
$\tilde{c}_t$	Expected cost of selected action at a time
$\mathcal{H}^\psi$	Logs in ψ-th interval
μ_b	Inter-arrival of change points
δ	Error range in the inter-arrival of change points
$\bar{t}(b_\upsilon)$	Mean occurrence time of a change point
H_0	Null hypothesis
H_1	Alternative hypothesis
ς	Change point moment
$\lambda(X)$	Likelihood ratio of the samples
$\mathcal{L}(D\|X_i)$	Likelihood function
$\hat{\mu}$	Mean of samples in H_0
$\hat{\mu}_0$	Mean of samples before change point in H_1
$\hat{\mu}_1$	Mean of samples after change point in H_1
σ^2	MLE for variance
α	Significance level
$\mathcal{D}$	Historical data in all intervals
π_0^ψ	Logging policy in ψ-th interval
π_w^ψ	Target policy in ψ-th interval
$V^*(\pi_w^\psi)$	Optimal value of the target policy in interval ψ
$\hat{V}_{\text{IPS}}(\pi_w^\psi)$	Estimated value of the target policy with IPS estimator
R_T^{Off}	Cumulative regret of off-policy approach

4.2 An On-line Learning Solution for Network Selection

The latency for a task depends on several parameters related to the EU and the BSs. An EU knows the task parameters such as the size and the required number of process operations. However, the traffic load in each network is unknown to the EU as it depends on the vehicles' arrivals and departures and the offloading demands in the corresponding coverage area. To this aim, in our latency minimization problem, we focus on task waiting time that depends on traffic load in each network. We utilize the single-player MAB model, which is suitable to solve the problems with limited information such as **P1**.

In a bandit model, an agent gambles on a machine with a finite set of arms. Upon pulling an arm, the agent receives some instantaneous reward from the reward generating process of the arm, which is not known a priori. Since the agent does not have sufficient knowledge, at each trial it might pull some inferior arm in terms of reward which results in some instantaneous regret. By pulling arms sequentially at different trials of the game, the agent aims at satisfying some optimality conditions [33]. Since our objective is to minimize the waiting time, we opt to use the notion of *cost* instead of reward. Therefore, the goal is to minimize the cost. In brief, in our model:

- The EU and the networks represent the agent and the arms, respectively.
- The instantaneous loss of pulling arm is the difference between the expected waiting time and the waiting time corresponding to the optimal arm.

At every round, the player selects an arm (i.e., a network), for offloading a task, observes its loss, and updates the estimation of its loss distribution. Each time a network is selected, the player observes the waiting time that is used for cost calculation. The objective is to minimize this loss brought by wrong network selection over time. We define the instantaneous cost function for choosing action m (network selection) at round t as

$$c_t^m = \left\{ T_{w_m}^l(\vartheta_t) \right\} \cdot \quad _{\{\mathbb{Q}_t^m = 1\}} \tag{9}$$

The offloading latency depends largely on the task queuing time; however, due to the dynamicity of a vehicular network such as vehicles' density, often no information is available about this variable. Moreover, the statistical characteristics of vehicles' arrival and density, as well as offloaded tasks generation distribution, change over time. Hence we make the following assumption:

Assumption 2. *For all BSs in network m, λ_m and μ_m do not remain constant; rather, they remain fixed over specific periods of time and change from one period to another.*

Due to the aforementioned assumption, the queue status of the BSs is piece-wise stationary, where the length of the period and the distribution are unknown. As mentioned in Section 3.1, for the BSs that belong to the same network type, the average rates of arrival and departure of the tasks are assumed to be the same, thereby have similar *average* queuing behavior. This assumption simplifies the learning process[2].

Network selection for task offloading with MAB is a sequential optimization problem. The previously offloaded tasks provide latency/cost information. However, this information may not be accurate due to insufficient sampling of each arm. Hence, an exploration-exploitation trade-off shall be addressed. One of the most seminal policies to address the exploration-exploitation trade-off is Upper the Confidence Bound (UCB) algorithm [39]. In the UCB algorithm, at every round of the game, an index is calculated for each arm corresponding to the average reward of pulling the arm in all previous rounds (the exploitation factor) and the tendency in pulling the arm for another round (the exploration factor). The UCB policy considers the entire reward history to calculate the arms' indexes; however, in a piece-wise stationary setting, the old observations are less important [40]. Hence, to calculate the arms' indexes, it

2. Note that, although the average rates are the same, the instantaneous values can differ.

would be beneficial to disregard the obsolete observations and consider only the last τ observations. In our vehicular scenario, we exploit *Sliding-Window UCB* (SW-UCB) [41] algorithm that uses the last τ observations for learning, as described in the following.

The number of times the mth arm has been selected during a window with length τ up to round t is given by

$$C_t^m(\tau) = \sum_{s=t-\tau+1}^{t} {}_{\{\mathbb{Q}_s^m=1\}}. \tag{10}$$

Let us define the total number of offloaded tasks by the EU, C_t, by round t to all the selected arms (networks) as

$$C_t = \sum_{m\in\Im_m} \sum_{s=1}^{t} {}_{\{\mathbb{Q}_s^m=1\}}. \tag{11}$$

Inspired by the SW-UCB, we define the cost index of pulling arm m at round t as

$$\hat{c}_t^m(\tau) = \bar{c}_t^m(\tau) - \beta\sqrt{\frac{\xi\log(\min\{C_t,\tau\})}{C_t^m(\tau)}}. \tag{12}$$

At every decision-making round, the agent pulls the arm with the minimum $\hat{c}_t^m(\tau)$. In (12), the first term on the right side corresponds to the exploitation factor, since $\bar{c}_t^m(\tau)$ is the average accumulated cost up to round t with window length τ. Formally,

$$\bar{c}_t^m(\tau) = \frac{1}{C_t^m(\tau)} \sum_{s=t-\tau+1}^{t} c_s^m. \tag{13}$$

The second term on the right side of (12) is the exploration factor, where ξ is a constant weight and β is an upper bound on exploration factor.

Let $L_t^*(m) = \min_m \mathbb{E}\left[c_t^m\right]$ represent the expected cost of offloading to the optimal network, while $L_t(m) = \mathbb{E}\left[\hat{c}_t^m(\tau)\right]$ denotes the expected cost of offloading to the mth network selected by the proposed MAB method. We define the cumulative regret during T rounds as

$$R_T^{\text{On}} = \sum_{t=1}^{T} L_t(m) - \sum_{t=1}^{T} L_t^*(m), \tag{14}$$

which is the expected loss of the algorithm compared with the optimal network selection.

4.3 An Off-Policy Learning Solution for Network Selection

In this section, we address the problem of network selection by proposing an off-policy learning method in a bandit setting. In off-policy learning, the goal is to estimate the value of a target policy exploiting a historical (or logging) policy. Off-policy learning can be seen as a parameterized policy such as weights in a neural network. In this setting, each request from the EU provides a context, based on which the system selects an action and incurs some cost. Such contextual-bandit data can be logged in large quantities and used for future purposes as training data [42]. Different from the on-line learning setting, off-policy learning is statistically more challenging since the collected logs are generated by a logging policy that differs from the current policy to be developed [43].

A large body of literature considers a stationary environment for off-policy evaluation; in this work, however, we consider a non-stationary environment. That is, in some time intervals both cost distribution and context remain stationary, though the distribution might change at some unknown time. Therefore, in this context, the problem of learning becomes two-fold: (i) estimating the occurrence time of the change points, and (ii) developing a target policy for each stationary interval based on the post-change distribution.

4.3.1 Change Point Detection

In an on-line setting, it is vital to detect a change as fast as possible while minimizing the rate of false alarms. In an off-line scenario, the goal is to identify the patterns or distribution of the change point occurrence based on the observed logs. For the formulated VEC problem, we propose a mixed off-line and on-line strategy for change point detection.

Let $\mathcal{B} = \{b_1,\ldots,b_\upsilon,\ldots,b_\Upsilon\}$ be the set of change points, and $\mathcal{Z} = \{z_1,\ldots,z_\psi,\ldots,z_\Psi\}$ the set of intervals, where $\Psi = \Upsilon + 1$. Each interval z_ψ has an unknown duration including ω rounds.

The input data consists of a finite streams of tuple $\left\langle \nu^\psi, \tilde{c}_t \right\rangle$ where ν^ψ is a context vector, $\nu^\psi = [\lambda_m^\psi, \mu_m^\psi]_{m=1}^M$, and λ_m^ψ and μ_m^ψ are the mth network task generation and service rate distribution parameters in the interval ψ. The context ν^ψ is a feature vector drawn according to some unknown distribution χ during each interval. If we assume that c_t^m in (9) can be characterized by an interval-dependent distribution[3] conditioned on ν^ψ and m, denoted by $D^\psi(c_t^m|\nu^\psi, m)$, we can define $\tilde{c}_t = \mathbb{E}\left[c_t^m\right]$ as the expected cost of the selected action at every time instant. Since an action implies the selection of a certain network, in the expectation we remove the dependency on the selected network.

Let $\mathcal{H}^\psi := (\nu^\psi, \tilde{c}_0, \ldots, \tilde{c}_\omega),\ \forall\omega \in z_\psi$ denote the logs or the history on the interval ψ for all ω rounds. We consider that the change points have an unknown distribution with expected inter-arrival of μ_b. We make the following assumption.

Assumption 3. *Given $\mathcal{H}^\psi$ for all intervals, the observations of the logs reveal μ_b by an error range of δ[4].*

We use the Likelihood Ratio Test (LRT) as an on-line procedure to detect the moment of the change by a sequential change point testing. Based on Assumption 3, the logs in the off-policy procedure allow us to obtain the range of the change point occurrence and to narrow down the period in which the LRT is performed. Fig. 2 depicts an example of the cost distribution of one network, which changes after each change point b_2 and b_3. The range of the change point occurrence where the LRT test is applied can also be seen in Fig. 2. The horizontal lines on c represent the expected cost of the network in the interval. The expected occurrence time of the υ-th change point can be written as

3. It should be noted that, in a non-stationary environment, both cost and context distribution change after each change points.

4. This assumption is realistic due to the recent advances in big data analysis that allows extracting and analyzing geometric and statistical patterns of massive size data sets [44].

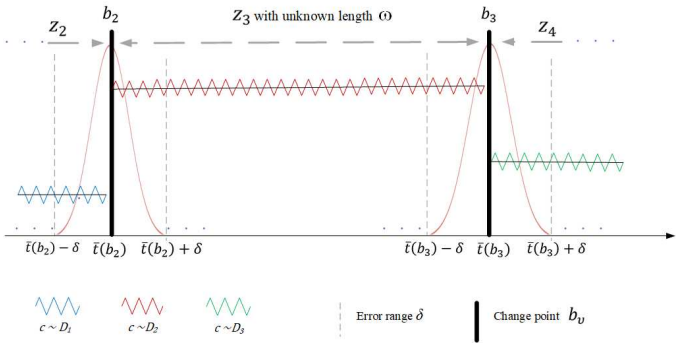


Fig. 2. Change point occurrence in a time series.

$\bar{t}(b_\upsilon) = \upsilon \cdot \mu_b$. Therefore, the occurrence range of the υ-th change point yields $[\bar{t}(b_\upsilon) - \delta \quad \bar{t}(b_\upsilon) + \delta]$. Exploiting the logs, there exists a point $\bar{t}(b_\upsilon) - \delta \in \mathbb{N}$ in the ψ-th interval such that $D_1 = D_2 = \ldots = D_{\bar{t}(b_\upsilon)-\delta}$, where D_ω is the cost distribution of the ω-th round belonging the the interval z_ψ. However, at some point in the future in the range of 2δ, the change point occurs and the LRT will be performed sequentially in such interval as also depicted in Fig. 2. We are interested to test the null hypothesis defined as

$$H_0 : D_1 = \cdots = D_{\varsigma-1} = D_\varsigma = \ldots, \tag{15}$$

against the alternative hypothesis defined as

$$H_1 : \exists \varsigma \in \mathbb{N} : D_1 = \cdots = D_{\varsigma-1} \neq D_\varsigma = D_{\varsigma+1} = \ldots, \tag{16}$$

by focusing on the occurrence of one change point in the alternative hypothesis, where ς is the change point. By assuming a truncated normal distribution in the range of change point occurrence [45], the likelihood ratio of the two hypotheses up to time $t' \in [\bar{t}(b_\upsilon) - \delta \quad \bar{t}(b_\upsilon) + \delta]$, where $t' \geq \varsigma$, for testing $H_0 : D \in \boldsymbol{D}_0$ against $H_1 : D \in \boldsymbol{D}$ yields

$$\Lambda(X) = \frac{\sup\limits_{D\in\boldsymbol{D}_0} \mathcal{L}(D|X_i)}{\sup\limits_{D\in\boldsymbol{D}} \mathcal{L}(D|X_i)} = \frac{\sup\limits_{\hat{\mu}} \prod\limits_{i=1}^{t'} f(\hat{\mu}, \sigma^2|X_i)}{\sup\limits_{\hat{\mu_0},\hat{\mu_1}} \prod\limits_{i=1}^{\varsigma} f(\hat{\mu}_0, \sigma^2|X_i) \cdot \prod\limits_{i=\varsigma+1}^{t'} f(\hat{\mu}_1, \sigma^2|X_i)} \tag{17}$$

where $f(\cdot|X_i)$ is the truncated normal probability density function, $\hat{\mu} = \frac{\sum_{i=1}^{t'} x_i}{t'}$, $\hat{\mu}_0 = \frac{\sum_{i=1}^{\varsigma} x_i}{\varsigma}$, $\hat{\mu}_1 = \frac{\sum_{i=\varsigma+1}^{t'} x_i}{(t'-\varsigma)}$, $\boldsymbol{D}$ is the parameter space, and $\boldsymbol{D}_0$ is a specified subset of it. Moreover, X_i are the samples taken from the logs.

Remark 1. *We apply the LRT to the samples taken from the mostly-selected network; however, in general, the test shall be performed for the samples of each of the networks.*

From $\frac{\partial \ln \mathcal{L}(\hat{\mu}_0,\hat{\mu}_1|X_i)}{\partial \sigma^2} = 0$, we obtain

$$\sigma^2 = \frac{\sum\limits_{i=1}^{\varsigma}(x_i - \hat{\mu}_0)^2 + \sum\limits_{i=\varsigma+1}^{t'}(x_i - \hat{\mu}_1)^2}{t'}, \tag{18}$$

where the most likely time of the change point of the variance[5] is $\varsigma = \text{argmin}_{5\leq\varsigma\leq t'-5}\{\sigma(\hat{\mu}_0, \hat{\mu}_1)\}$. Through some algebraic steps, (17) can be rewritten as

$$\Lambda(X) = \left(\frac{\sum\limits_{i=1}^{\varsigma}(x_i - \hat{\mu}_0)^2 + \sum\limits_{i=\varsigma+1}^{t'}(x_i - \hat{\mu}_1)^2}{\sum\limits_{i=1}^{t'}(x_i - \hat{\mu})^2} \right)^{t'/2}. \tag{19}$$

The LRT for testing H_0 against H_1 has the critical region of the form $\{x : \Lambda(x) \leq k\}$, where k is a real number in range [0, 1]. The test will be at significance level α if k satisfies

$$\sup\{P(\Lambda(x) \leq k; D \in \boldsymbol{D}_0)\} = \alpha.$$

That is, low values of LRT imply that the observed result is less likely to occur under H_0 than H_1; therefore, the null hypothesis shall be rejected.

4.3.2 Off-Policy Learning

The historical data $\mathcal{D} := (\mathcal{H}^\psi, \pi_0^\psi)_{\psi=1}^{\Psi}$ is given by executing the logging policy π_0^ψ on interval ψ. Our goal is to develop a target policy π_w^ψ for each interval exploiting the historical data $\mathcal{D}$. Both logging and target policies map the context to the network with the highest probability distribution. Given n independent and identically-distributed samples, we wish to compute the value of the policy π_w^ψ on interval ψ as

$$\begin{aligned} V^*(\pi_w^\psi) &= \mathbb{E}\left[\sum_{t=0}^{\omega} c_t^m \middle| \pi_w^\psi, D^\psi, \nu^\psi\right] = \mathbb{E}_{\pi_w^\psi}\left[c_t^m\right] \\ &= \mathbb{E}_{\nu^\psi\sim\chi}\mathbb{E}_{m\sim\pi_w^\psi(\cdot|\nu)}\mathbb{E}_{c\sim D^\psi(\cdot|m,\nu)}\left[c_t^m\right]. \end{aligned} \tag{20}$$

To estimate the value of policy π_w^ψ on interval ψ with any estimator we have the following

$$\hat{V}(\pi_w^\psi|\mathcal{D}) := \hat{V}(\pi_w^\psi|\mathcal{H}^\psi, \pi_0^\psi). \tag{21}$$

A widely-used off-policy estimator is Inverse Propensity Score (IPS) [47], [48] that is used in this paper to evaluate the value of a policy. In a general sense, given logging policy π_0, target policy π_w, the logged data, and their distributions, the value of the target policy π_w based on IPS estimator is

$$\begin{aligned} \hat{V}_{\text{IPS}}(\pi_w) &= \frac{1}{T}\sum_{t=1}^{T}\sum_{m}\pi_w(m|\nu_t)\frac{\{m_t=m\}}{\pi_0(m_t|\nu_t)}\tilde{c}_t \\ &= \frac{1}{T}\sum_{t=1}^{T}\frac{\pi_w(m_t|\nu_t)}{\pi_0(m_t|\nu_t)}\tilde{c}_t, \end{aligned}$$

where $\tilde{c}_t$, as defined earlier, is the expected cost of the selected network.

Definition 1. The logging policy π_0 is said to have full support for π_w when $\pi_0(m|\nu) > 0 \quad \forall \nu, m$ for which $\pi_w(m|\nu) > 0$.

In a stationary setting, to ensure that $\hat{V}_{\text{IPS}}(\pi_w)$ is an unbiased estimate of $V^*(\pi_w)$, π_0 should have full support for π_w, i.e. assigning non-zero probabilities to every action

5. To avoid misbehavior of the likelihood function, ς shall be at least 5 observations apart from the first and the last values in the series [46].

in every context [49]. According to (21), we can write $\hat{V}_{\text{IPS}}(\pi_w^{\psi}|\mathcal{D}) = \hat{V}_{\text{IPS}}(\pi_w^{\psi}|\mathcal{H}^{\psi}, \pi_0^{\psi})$. We define

$$\hat{V}_{\text{IPS}}(\pi_w^{\psi}|\mathcal{H}^{\psi}, \pi_0^{\psi}) = \sum_{t=1}^{\omega} \frac{\pi_w^{\psi}(m_t|\nu_t)}{\pi_0^{\psi}(m_t|\nu_t)} \tilde{c}_t. \tag{22}$$

Therefore, the optimal policy of the IPS estimator in each interval can be written as

$$\underset{\pi_w^{\psi}}{\operatorname{argmin}}\, \hat{V}_{\text{IPS}}(\pi_w^{\psi}|\mathcal{H}^{\psi}, \pi_0^{\psi}). \tag{23}$$

To estimate the value of the policy π_w over all intervals in our non-stationary environment, we define $\hat{V}_{\text{IPS}}(\pi_w)$ as

$$\hat{V}_{\text{IPS}}(\pi_w) = \frac{1}{\Psi} \sum_{\psi=1}^{\Psi} \hat{V}_{\text{IPS}}(\pi_w^{\psi}), \tag{24}$$

which is the average of the IPS estimator from each interval. Moreover, we define the regret of the off-policy approach as

$$R_T^{\text{Off}} = \sum_{\psi=1}^{\Psi} \hat{V}_{\text{IPS}}(\pi_w^{\psi}) - \sum_{\psi=1}^{\Psi} V^*(\pi_w^{\psi}), \tag{25}$$

which is the expected loss of the algorithm compared with the optimal network selection. The two-step off-policy approach is presented in **Algorithm 1**, where given the data set $\mathcal{D}$, the policy for the network selection in each interval is developed based on the on-line SW-UCB approach.

Algorithm 1 The Off-Policy Approach

Input: $\mathcal{D}$
Output: $\pi_w^{\psi} \quad \forall z_{\psi} \in \mathcal{Z}$
$\upsilon = 0$, $\psi = 1$
for t=1: T **do**
  Calculate:
  the estimated change point moment (ς),
  mean before the estimated change point ($\hat{\mu}_0$),
  mean after the estimated change point ($\hat{\mu}_1$),
  and mean of null hypothesis ($\hat{\mu}$)
  Estimate the LRT using (19)
  if $\Lambda(X) < k$ **then**
    Extract $\mathcal{H}^{\psi}$, π_0^{ψ} from $\mathcal{D}$
    Estimate π_w^{ψ} using (23)
    $\upsilon = \upsilon + 1$, $\psi = \psi + 1$
  end if
end for

4.3.3 Overview of the Off-Policy Procedure

Concerning the off-policy learning, we use the proposed on-line solution as the logging policy (π_0) to develop the target policy (π_w). That is, the *SW-UCB* is the policy based on which the logs are obtained[6]. As discussed in Section 4.3.2, given the logged data, the target policy π_w for each interval is the solution of the following optimization problem:

$$\begin{aligned}\mathbf{P2}: \underset{\pi_w^{\Psi}}{\operatorname{argmin}} \Bigg\{ & \pi_w^{\Psi}(m|\nu) \cdot \quad_{\{m=1\}} \cdot \sum_{t=1}^{\omega} \frac{\{m_t=1\}}{\pi_0^{\Psi}(m_t|\nu_t)} \cdot \tilde{c}_t \\ & + \pi_w^{\Psi}(m|\nu) \cdot \quad_{\{m=2\}} \cdot \sum_{t=1}^{\omega} \frac{\{m_t=2\}}{\pi_0^{\Psi}(m_t|\nu_t)} \cdot \tilde{c}_t \\ & + \pi_w^{\Psi}(m|\nu) \cdot \quad_{\{m=3\}} \cdot \sum_{t=1}^{\omega} \frac{\{m_t=3\}}{\pi_0^{\Psi}(m_t|\nu_t)} \cdot \tilde{c}_t \Bigg\}\end{aligned} \tag{26}$$

6. In our numerical analysis, we simulate *SW-UCB* policy as π_0 to log the network selection outcomes as the data set $\mathcal{D}$.

subject to

$$\begin{aligned}\mathbf{C2}: & \pi_w^{\Psi}(m|\nu) \cdot \quad_{\{m=1\}} + \pi_w^{\Psi}(m|\nu) \cdot \quad_{\{m=2\}} \\ & + \pi_w^{\Psi}(m|\nu) \cdot \quad_{\{m=3\}} = 1 \end{aligned} \tag{27}$$

$$\mathbf{C3}: \pi_w^{\Psi}(m|\nu) > 0 \quad \forall m. \tag{28}$$

Constraint (27) guarantees that the sum of the probabilities equals to one. Constraint (28) assigns non-zero probability of selection to each network according to Definition 1. The optimization problem **P2** can be simply solved using standard solvers such as *CPLEX* with the execution time being less than 0.1 second on a modest hardware. Moreover, after detecting a change point, the optimization problem is solved only once at the beginning of the triggered interval, so that its effect on the overall latency is negligible.

Remark 2. *The performance of the π_w depends on several parameters such as number of arms (networks) and the quality of π_0. Indeed, large number of arms or weak performance of π_0 increases error probability in π_w.*

5 BS Selection and Relaying Mechanism

5.1 BS Selection

Once the least congested network type is identified, one of the BSs in the network should be selected for offloading. However, as mentioned before, the BSs of the same network type have on average the same waiting time. Moreover, small waiting time is not the sufficient condition for a BS to be the best among the available ones, mainly due to the effect of some other factors such as the sojourn time. To guarantee a successful offloading, i.e. the reception of results by the vehicle, we consider the sojourn distance in the coverage of the BS as a parameter when selecting a BS inside the previously-identified least congested network.

As shown in Fig. 3, in a task offloading procedure in our framework, there can be eight offloading cases depending on the locations of the devices. Considering all offloading cases for the ith EU, the sojourn/remaining distance[7] before going out of the coverage of the jth BS at time instant t is equal to [50]

$$\Delta_t^{i,m_j} = \begin{cases} |x_{m_j} - x_{i,t}| + x' & \text{cases 1,2,5,6 in Fig. 3} \\ x' - |x_{m_j} - x_{i,t}| & \text{cases 3,4,7,8 in Fig. 3,} \end{cases} \tag{29}$$

where

$$x' = \sqrt{R_m^2 - h_{m_j}^2 - |y_{m_j} - y_{i,t}|^2} \tag{30}$$

is a distance inside the BS coverage as depicted in Fig. 1. As a first step for the BS selection procedure, the EU identifies the BSs in the selected network, as long as it is within their coverage area, as potential candidates for offloading. Hence, we define the set of candidates, $\Im_t^F \subset \Im_j^m$, in the selected network m that are available for the EU for computation offloading at time instant t as

$$\Im_t^F = \Big\{ N_{m_j} | d_t^{i,m_j} < R_m, \mathbb{Q}_t^m = 1 \Big\}, \tag{31}$$

7. The sojourn distance can be calculated by the EU at any time, since it only requires the knowledge about the locations of the BSs which are fixed.

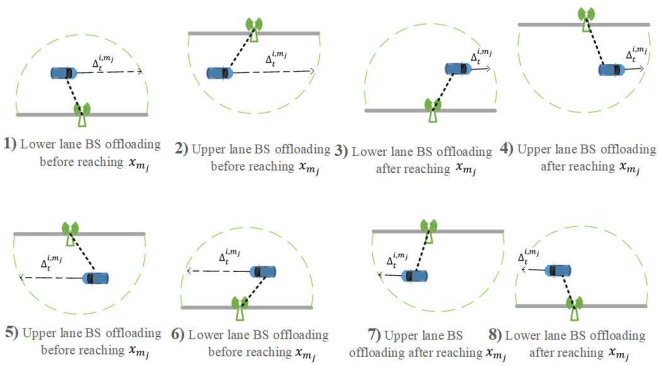


Fig. 3. Offloading cases considering the locations of devices.

where d_t^{i,m_j} is the Euclidean distance between the EU and the jth BS of the mth network as defined in (5). Moreover, R_m is the coverage area of every BS in network type m. Considering the sojourn distance for the BS selection phase, the best BS is selected as

$$\underset{N_{m_j}\in\Im_t^F}{\operatorname{argmax}}\left\{\Delta_t^{i,m_j}\right\}. \tag{32}$$

Remark 3. *In case the number of EUs accessing each BS is restricted to some value $\bar{N}$; That is, there is a constraint* $\mathbf{C2}$: $|\Im_{m_j,t}^i| \leq \bar{N}$, *where* $\Im_{m_j,t}^i = \{u_i|\mathbb{B}_{m_j}^i = 1\}$ *and* $\mathbb{B}_{m_j}^i$ *is an indicator function which returns 1 if EU i offloads to BS m_j. Let ϱ be an indicator function that returns 1 if* $\mathbf{C2}$ *holds and zero otherwise. In this case, the BS is selected as*

$$\underset{N_{m_j}\in\Im_t^F}{\operatorname{argmax}}\left\{\Delta_t^{i,m_j}\right\}\cdot \quad \{\varrho=1\}. \tag{33}$$

5.2 Relaying Mechanism

In VEC, the probability of task loss (or task outage) corresponds to the probability that the EU does not receive an offloaded task in due time, which occurs as a result of the EU mobility. We aim at minimizing the probability of task loss. First, we note that the time that the ith EU remains in the coverage area of the jth BS (i.e., sojourn time) yields

$$\widetilde{T}_t^{i,m_j} = \frac{\Delta_t^{i,m_j}}{v_i}. \tag{34}$$

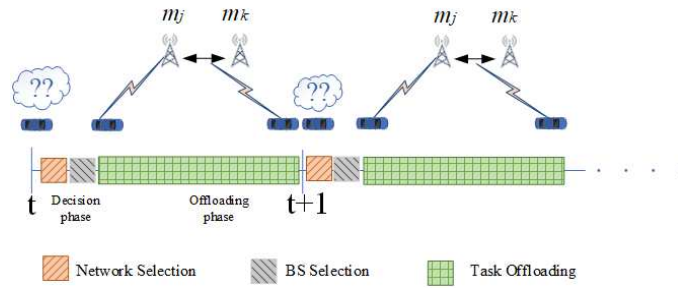
Moreover, the overall offloading time is given by

$$T_{l,t}^{\text{off},m_j} = \frac{\rho_l^{\text{up}}}{r_{m_j,t}^{\text{up}}} + \frac{O\cdot\rho_l^{\text{up}}}{\eta_{m_j}} + T_{w_m}^l(\vartheta_t^m) + \frac{\rho_l^{\text{dl}}}{r_{m_j,t}^{\text{dl}}}. \tag{35}$$

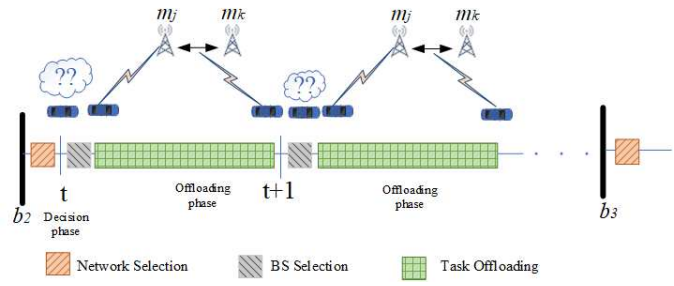
For the lth task generated at time instant t, the task loss can be formalized as

$$\Omega_{l,t}^{m_j} = \begin{cases} 1 & \text{if } \widetilde{T}_t^{i,m_j} < T_{l,t}^{\text{off},m_j} \\ 0 & \text{if } \widetilde{T}_t^{i,m_j} \geq T_{l,t}^{\text{off},m_j} \end{cases}. \tag{36}$$

In words, a task loss occurs if the total offloading latency is larger than the EU sojourn time within the coverage area of the serving BS. The EU aims at minimizing the task loss during the time horizon T, i.e., minimize $\sum_{t=1}^{T}\Omega_{l,t}^{m_j}$. Even for the best BS w.r.t. the sojourn distance, the overall offloading latency might be more than the sojourn time. To address this challenge, we develop a relaying mechanism.



(a) The time-scale of the two phases for the on-line approach.



(b) The time-scale of the two phases for the off-policy approach.

Fig. 4. The time-scale of the two phases for the two learning approaches.

Let the *original BS* refer to the BS selected for task offloading. The *neighboring BS* is a BS towards which (i.e. its coverage area) the vehicle moves. A relaying mechanism allows a vehicle to collect the task result from the neighboring BS, if receiving the result from the original BS is not possible. That is, if the BSs can communicate with each other, the task is relayed through the backhaul network [20]. Then the relayed offloading time yields

$$\hat{T}_{l,t}^{\text{off},m_j} = \frac{\rho_l^{\text{up}}}{r_{m_j,t}^{\text{up}}} + \frac{O\cdot\rho_l^{\text{up}}}{\eta_{m_j}} + T_{w_m}^l(\vartheta_t^m) + T_l^{\text{trans},Op} + \frac{\rho_l^{\text{dl}}}{r_{m_k,t}^{\text{dl}}}, \tag{37}$$

where $T_l^{\text{trans,Op}}$ is the transmission time for relaying through backhaul, and $\frac{\rho_l^{\text{dl}}}{r_{m_k,t}^{\text{dl}}}$ the reception time from the destination BS. We then redefine the outage as

$$\hat{\Omega}_{l,t}^{m_j} = \begin{cases} 1 & \text{if } \widetilde{T}_t^{i,m_j} < \frac{\rho_l^{\text{up}}}{r_{m_j,t}^{\text{up}}} \\ 0 & \text{if } \widetilde{T}_t^{i,m_j} \geq \frac{\rho_l^{\text{up}}}{r_{m_j,t}^{\text{up}}} \end{cases}. \tag{38}$$

This implies that when employing a relaying mechanism, a task outage occurs only if the transmission time to the original BS is not sufficient.

6 Performance Evaluation

We evaluate the performances of the proposed learning methods by simulations.

6.1 Time-Scale of Operations of the Learning Algorithms

In order to have a better understanding of the scenario considered for simulation results, we elaborate here the time-scale of the two proposed learning algorithms, each composed of a decision-making and an offloading phase.

As illustrated in Fig. 4, in the first phase, the EU decides where to offload. This decision phase includes network selection, as explained in Section 4, and BS selection, as explained in Section 5.1. Afterward, the user offloads the task to the selected BS and later receives the result either from the *original* BS or the *neighboring* BS, in case of relaying, as explained in Section 5.2.

The on-line approach includes the network selection and BS selection at every round of decision-making, i.e., upon generation of a task, as illustrated in Fig. 4a. After selecting the BS, the offloading starts. The off-policy approach, in contrast, performs the network selection only upon detecting a change point, as explained in Section 4.3.2[8]. However, upon generation of a task at every round, only the BS selection is performed. Similar to the on-line approach, the offloading phase starts after selecting the BS. It should be noted that the learning solutions do not incur any signaling overhead, since there is no message exchange between the vehicles and the BSs in the learning procedure.

6.2 Simulation Parameters

To evaluate the proposed offloading schemes, we perform numerical analysis. Table 2 summarizes the simulation parameters. The vehicle has a random location in the area. There are three networks (e.g. Macro, Micro, and Pico cellular networks) available in the area. The Macro and Micro BSs are placed in the upper and lower side of the road and the Pico BSs in the upper, lower and middle of the road all following uniform distribution. By this placement of BSs, at least one network is available to serve the vehicle at all time. There are 25000 simulation rounds out of which we consider only the rounds in which all the three network types are available to guarantee a fair comparison.

Table 3 presents the traffic of each network in different intervals. A vehicle generates a task sequentially. We consider two applications: (i) A processing application generating tasks that requires a higher number of processing operations (e.g. image processing); (ii) A collecting application that requires a lower number of processing operations, (e.g. sensor data analysis). In Table 4, we summarize the numerical values, expressed in terms of Floating Point Operations (FLOP) per task data size [51].

6.3 Impact of UCB Parameters on the On-line Approach

We first investigate the impact of parameters on the performance of *SW-UCB*. Fig. 5 shows the impact of window length on both number of sub-optimal network selection and the average regret. Sub-optimal network selection is mainly affected by the window length. With a small window length the change point is detected faster, whereas a large window size might result in detection delay. Nonetheless, if the window is too small, there might not exist sufficient historical data for optimal decision-making during the interval between two change points. Based on the experiments, we select the window length as $\tau = 100$ rounds for rest of the simulation as it results in the best network selection, thereby the lowest average regret.

8. Change point detection algorithm is run within the range of 2δ as explained in Section 4.3.1.

TABLE 2
Simulation Parameters

Parameter	Value
Height of Ma-BS	20 m [52]
Height of Mi-BS	6 m [52]
Height of Pi-BS	3 m [52]
Path-loss attenuation factor (β_t^{i,m_j})	$140.7 + 36.7 \log_{10}\left(\frac{d_t^{i,m_j}}{1000}\right)$ [53]
Bandwidth (B_m)	10 MHz [28]
Noise power density (ζ)	$10^{-13} B_m$ [28]
Ma-BS coverage range	500 m [52]
Mi-BS coverage range	200 m [52]
Pi-BS coverage range	100 m [52]
EU computational power (η_i)	15 G FLOPS
Ma computational power	$4 * \eta_i$
Mi computational power	$3 * \eta_i$
Pi computational power	$2 * \eta_i$
Significance level (α)	0.05
EU velocity (v_i)	10-20 meters/second
Error range in LRT (δ)	500 rounds
Inter-arrival of change points (μ_b)	5000 rounds

TABLE 3
MAB Setting

	Macro Network	Micro Network	Pico Network
Intervals	t=1∼ 5000 t=5000∼ 10000 t=10000∼ 15000 t=15000∼ 20000 t=20000∼ 25000	t=1∼ 5000 t=5000∼ 10000 t=10000∼ 15000 t=15000∼ 20000 t=20000∼ 25000	t=1∼ 5000 t=5000∼ 10000 t=10000∼ 15000 t=15000∼ 20000 t=20000∼ 25000
Traffic model parameters	λ=7 , μ=0.4 λ=3 , μ=0.3 λ=10 , μ=0.5 λ=3 , μ=0.3 λ=7 , μ=0.4	λ=3 , μ=0.3 λ=10 , μ=0.5 λ=7 , μ=0.4 λ=10 , μ=0.5 λ=10 , μ=0.5	λ=10 , μ=0.5 λ=7 , μ=0.4 λ=3 , μ=0.3 λ=7 , μ=0.4 λ=3 , μ=0.3
Expected cost	$c_t^m = 2.8$ $c_t^m = 0.9$ $c_t^m = 5$ $c_t^m = 0.9$ $c_t^m = 2.8$	$c_t^m = 0.9$ $c_t^m = 5$ $c_t^m = 2.8$ $c_t^m = 5$ $c_t^m = 5$	$c_t^m = 5$ $c_t^m = 2.8$ $c_t^m = 0.9$ $c_t^m = 2.8$ $c_t^m = 0.9$

We also study the impact of β, i.e. the coefficient of the exploration factor in the UCB index, on the average regret. Fig. 6 depicts the results. We select $\beta = 0.8$ for the rest of the simulation due to its superior performance. Furthermore, the effect of ξ on the average regret is shown in Fig. 7. From the figure, the value of $\xi = 0.2$ has the best performance.

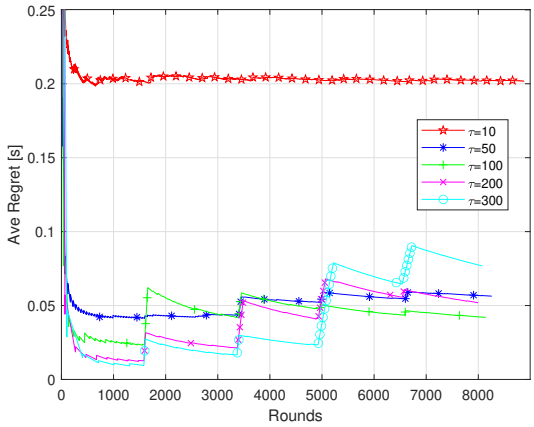
6.4 Comparison With Other Approaches

We compare the performance of *SW-UCB* and *Off-policy* solutions with the following benchmarks:

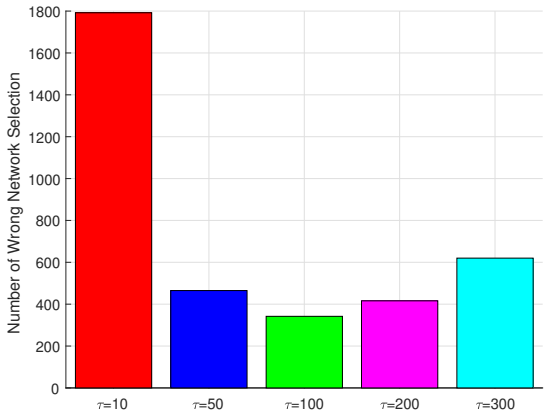
- *UCB*: The seminal upper-confidence bound policy [54];
- *ϵ-greedy*: At each round t, it selects a random arm with probability $\epsilon = 1/t$ and the best arm so far with probability 1-ϵ.

TABLE 4
Task Parameters

Task Parameter	Value
Task size (ρ_l^{up})	[1 5] MB
Offloaded to downloaded portion	5
Processing application operations	10 G FLOP per MB
Collecting application operations	1 G FLOP per MB



(a) Impact of τ on average regret.



(b) Impact of τ on sub-optimal network selection.

Fig. 5. Impact of the window length τ.

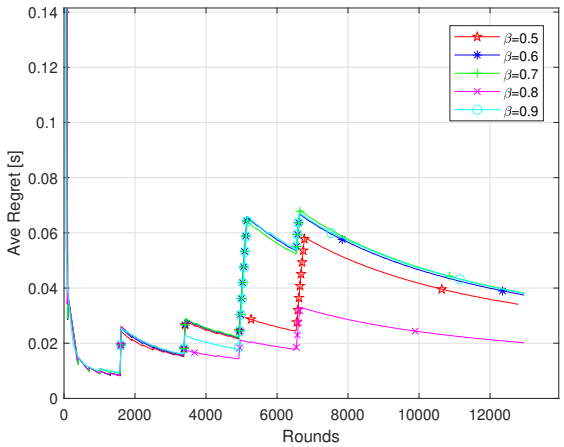


Fig. 6. Impact of β on average regret.

- *Random*: At each round, it selects an arm uniformly at random.
- *TOD*: It selects the network based on an approach belonging to the UCB family, namely, *discounted-UCB*. This approach assigns an index to each choice at every round of decision-making. In calculating the index, recent observations play a significant role as they receive high weights, and the effect of observations diminishes over time [55].

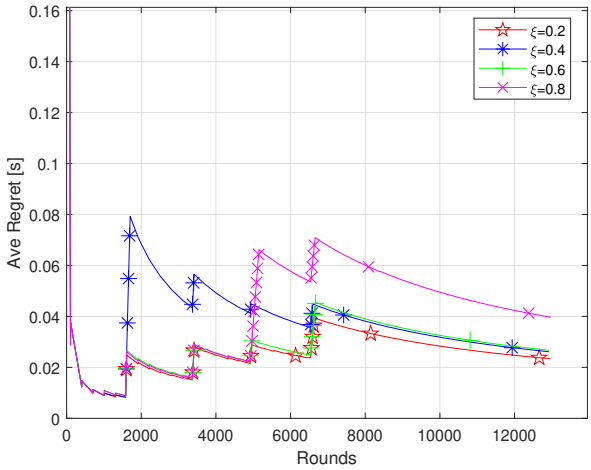


Fig. 7. Impact of ξ on average regret.

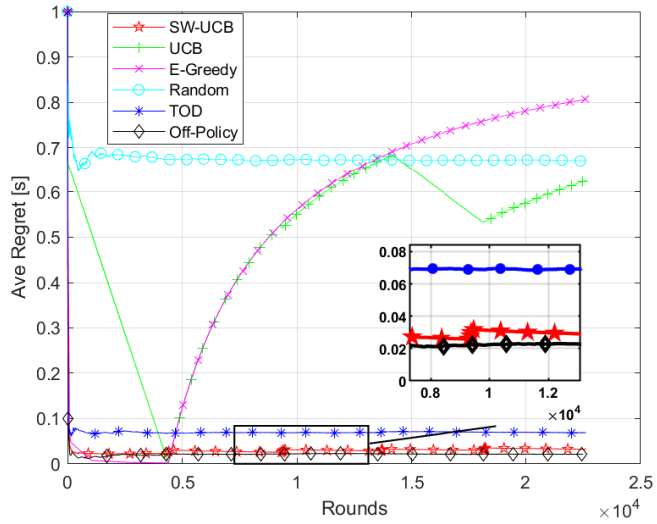


Fig. 8. Average regret.

Fig. 8 illustrates the performance of different approaches in terms of average regret. The proposed *SW-UCB* approach shows the best performance compared to other benchmarks. *TOD* approach is the closest to our proposed solutions. *ϵ-greedy* and *UCB* exhibit similar performance which is inferior in comparison with other methods. *Random* selection leads to a severe sub-optimal network selection. *Off-policy* approach has the lowest regret due to change point detection mechanism and the proposed network selection method.

Fig. 9 represents the network selection of the agent in different intervals while using different approaches. The parameters regarding the interval length and the expected cost in each interval are gathered in Table 3. In the first interval, four approaches, namely *SW-UCB*, *UCB*, *ϵ-greedy*, and *Off-Policy* show similar performance as they continue pulling

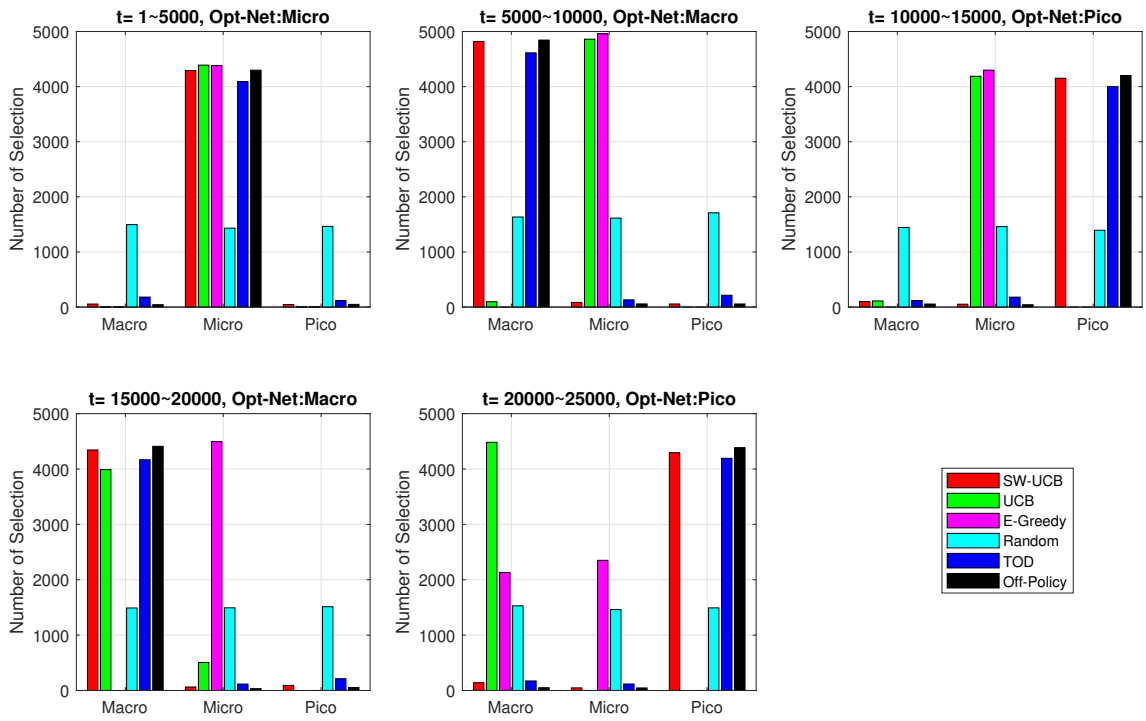


Fig. 9. Network selection in each interval.

the optimal arm. Nonetheless, after the change point, i.e. when the optimal network changes to Macro, their performances become different. The *Off-Policy* approach shows the best performance due to the LRT for change point detection and the developed π_w. Moreover, *SW-UCB* is able to detect the change and thus performs well. *TOD* has the closest performance to *SW-UCB* due to using the discounted-UCB algorithm. The *UCB* method considers the entire history for decision-making; therefore, it requires a long time to adapt to the change. Similarly, *ϵ-greedy* only exploits the historical knowledge and thus continues selecting the Micro network (exploration loses its color by time). *Random* approach does not take the changes in cost distribution into account and hence has the worst performance. All approaches show the same behavior in the other intervals.

Fig. 10 depicts the average regret, which is determined by the frequency of sub-optimal network selection and also the selected network. The average regret of *SW-UCB* and *Off-Policy* approaches are the smallest compared to the other methods. Fig. 11 depicts the average waiting time each task experiences at the selected BS. Clearly, by selecting the network with the lowest congestion in each interval, the tasks suffer lower waiting time. The proposed *SW-UCB* and *Off-Policy* approaches have the lowest waiting time for the tasks, which is the result of the optimal network selection in different congestion patterns. Finally, Fig. 12 shows that the relaying mechanism improves the performance in terms of the task loss and brings it to a value close to zero. Indeed, the residual loss in the absence of a relaying mechanism is mainly due to the small sojourn time that is insufficient even for transmission of the task to the original BS. The relaying mechanism mitigates this task loss.

7 Conclusion

We have studied a task offloading problem in a VEC scenario where several wireless access networks with dynamic traffic patterns co-exist. We have proposed a multi-arm bandit approach, namely, the *SW-UCB* approach, to solve the network selection problem. We have also proposed an off-policy approach to detect the change points and select the best network. Moreover, we have developed a BS selection and a relaying mechanism that reduces the waiting time and task loss. Numerical results have demonstrated the effectiveness of the proposed approaches in selecting the least congested network and adapting to the changes in the network traffic. The proposed *SW-UCB* has a lower average regret and lower task latency than other benchmarks. Moreover, the *Off-Policy* approach has the lowest regret over rounds and average task waiting time than all the other approaches.

As a future work, we plan to extend the single-agent MAB scenario to a multi-agent setting. In such a scenario, the traffic in each network depends on the joint decisions of agents that complicates the problem. Moreover, we would like to consider the problem of cost optimization of the system in terms of number/density of activated BSs required to maintain the latency for task offloading to an acceptable level. Furthermore, we plan to study the impact of energy consumption on the vehicular environment and consider a joint latency and energy consumption optimization.

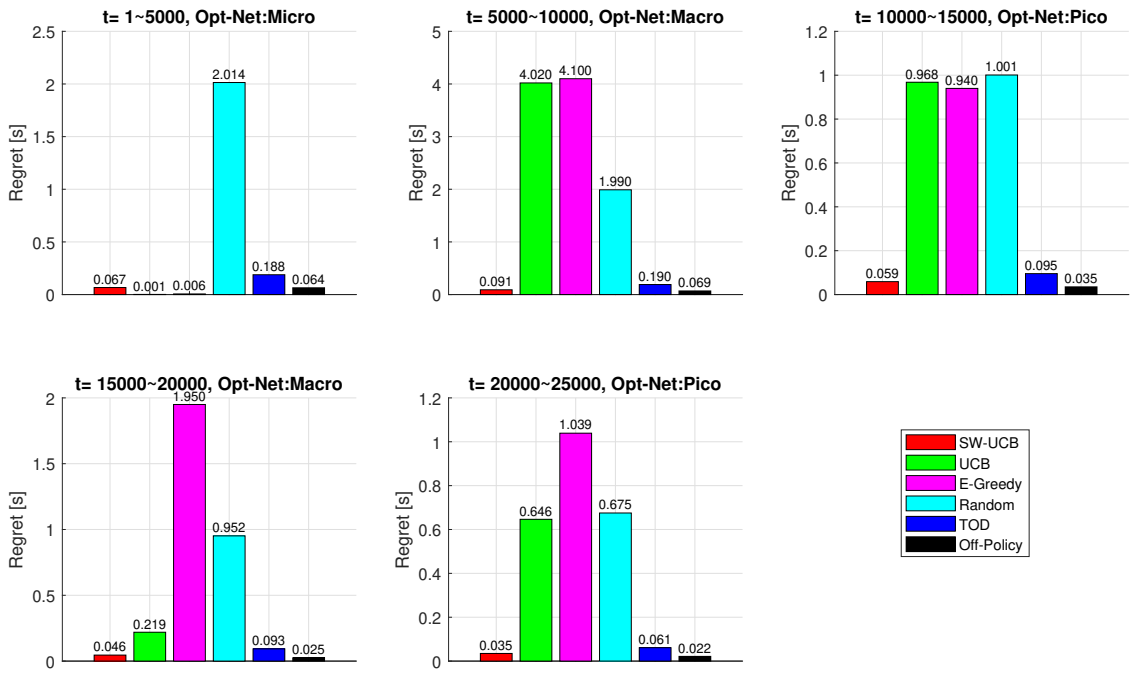


Fig. 10. Average regret in each interval.

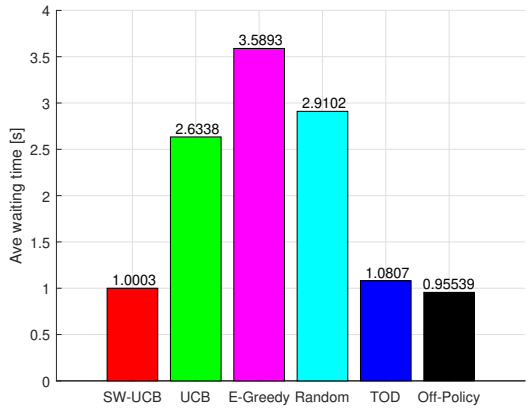


Fig. 11. Average waiting time for an offloaded task.

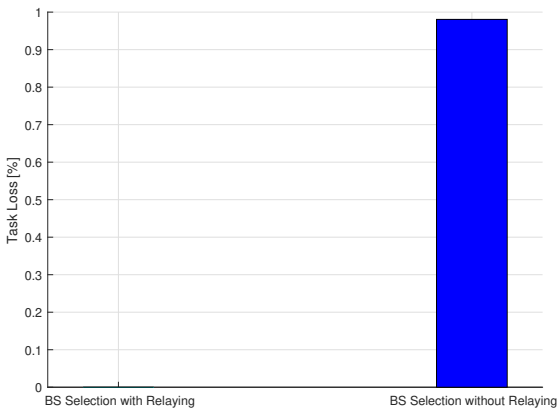


Fig. 12. Task loss with and without relaying mechanism.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[2] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, Thirdquarter 2017.

[3] F. Giust, X. Costa-Perez, and A. Reznik, "Multi-access edge computing: An overview of ETSI MEC ISG," *IEEE 5G Tech Focus*, vol. 1, no. 4, Dec. 2017. [Online]. Available: https://futurenetworks.ieee.org/tech-focus/december-2017/multi-access-edge-computing-overview-of-etsi

[4] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[5] S. Bitam, A. Mellouk, and S. Zeadally, "VANET-cloud: a generic cloud computing model for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 96–102, Feb. 2015.

[6] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a resource (VaaR)," *IEEE Netw.*, vol. 29, no. 1, pp. 12–17, Jan.-Feb. 2015.

[7] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81–93, Jan. 2015.

[8] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[9] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[10] O. Boxma and B. Zwart, "Tails in scheduling," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 4, pp. 13–20, Mar. 2007.

[11] E. P. Xing, Q. Ho, W. Dai, J. K. Kim, J. Wei, S. Lee, X. Zheng, P. Xie, A. Kumar, and Y. Yu, "Petuum: A new platform for distributed machine learning on big data," *IEEE Trans. Big Data*, vol. 1, no. 2, pp. 49–67, Jun. 2015.

[12] R. Jagerman, I. Markov, and M. de Rijke, "When people change their mind: Off-policy evaluation in non-stationary recommendation environments," in *WSDM'19 - Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, Melbourne VIC, Australia, Jan. 2019, pp. 447–455.

[13] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[14] J. Du, L. Zhao, X. Chu, F. R. Yu, J. Feng, and C. I, "Enabling low-latency applications in LTE-A based mixed fog/cloud computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1757–1771, Feb. 2019.

[15] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "A control and data plane split approach for partial offloading in mobile fog networks," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain, Apr. 2018.

[16] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 1459–1467.

[17] H. Yu, M. H. Cheung, L. Huang, and J. Huang, "Power-delay tradeoff with predictive scheduling in integrated cellular and wi-fi networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 735–742, 2016.

[18] J. Wang, J. Peng, Y. Wei, D. Liu, and J. Fu, "Adaptive application offloading decision and transmission scheduling for mobile cloud computing," in *2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016.

[19] D. Wang, Z. Liu, X. Wang, and Y. Lan, "Mobility-aware task offloading and migration schemes in fog computing networks," *IEEE Access*, vol. 7, pp. 43 356–43 368, 2019.

[20] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.

[21] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *2017 IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[22] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[23] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1079–1092, Feb. 2019.

[24] K. Zheng, F. Liu, Q. Zheng, W. Xiang, and W. Wang, "A graph-based cooperative scheduling scheme for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1450–1458, May 2013.

[25] L. Quan, Z. Wang, and F. Ren, "A novel two-layered reinforcement learning for task offloading with tradeoff between physical machine utilization rate and delay," *Future Internet*, vol. 10, no. 7, Jul. 2018.

[26] L. Mai, N.-N. Dao, and M. Park, "Real-time task assignment approach leveraging reinforcement learning with evolution strategies for long-term latency minimization in fog computing," *Sensors*, vol. 18, no. 9, Sep. 2018.

[27] F. Mashhadi, S. A. S. Monroy, A. Bozorgchenani, and D. Tarchi, "Optimal auction for delay and energy constrained task offloading in mobile edge computing," *Computer Networks*, vol. 183, Dec. 2020.

[28] Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Apr. 2019.

[29] H. Wu, X. Guo, and X. Liu, "Adaptive exploration-exploitation tradeoff for opportunistic bandits," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, Jul. 2018, pp. 5306–5314.

[30] X. Zhang, R. Zhou, Z. Zhou, J. C. C. Lui, and Z. Li, "An online learning-based task offloading framework for 5G small cell networks," in *49th International Conference on Parallel Processing - ICPP*, ser. ICPP '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3404397.3404417

[31] Z. Sun and M. R. Nakhai, "An online learning algorithm for distributed task offloading in multi-access edge computing," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3090–3102, 2020.

[32] S. Ghoorchian and S. Maghsudi, "Multi-armed bandit for energy-efficient and delay-sensitive edge computing in dynamic networks with uncertainty," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2020.

[33] S. Maghsudi and D. Niyato, "On power-efficient planning in dynamic small cell networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 304–307, Jun. 2018.

[34] B. Nikfar, S. Maghsudi, and A. J. H. Vinck, "Multi-armed bandit channel selection for power line communication," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Miami, FL, USA, Nov. 2015, pp. 19–24.

[35] S. Maghsudi and M. van der Schaar, "Distributed task management in cyber-physical systems: How to cooperate under uncertainty?" *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 1, pp. 165–180, 2019.

[36] Y. Sun, L. Xu, Y. Tang, and W. Zhuang, "Traffic offloading for online video service in vehicular networks: A cooperative approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7630–7642, Aug. 2018.

[37] Y. Pei, H. Tian, T. Wu, and Y. Wang, "Frequency planning scheme based on interference coordination for ofdm-relay systems," *IEEE Communications Letters*, vol. 15, no. 1, pp. 13–15, 2011.

[38] B. Haji and S. Ross, "Minimizing expected discounted cost in a queueing loss model with discriminating arrivals," *European Journal of Operational Research*, vol. 282, no. 2, pp. 593 – 601, Apr. 2020.

[39] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, May 2002.

[40] C. Hartland, S. Gelly, N. Baskiotis, O. Teytaud, and M. Sebag, "Multi-armed Bandit, Dynamic Environments and Meta-Bandits," Nov. 2006, working paper or preprint. [Online]. Available: https://hal.inria.fr/hal-00113668/

[41] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *Algorithmic Learning Theory*, J. Kivinen, C. Szepesvári, E. Ukkonen, and T. Zeugmann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 174–188.

[42] T. Joachims, A. Swaminathan, and T. Schnabel, "Unbiased learning-to-rank with biased feedback," in *WSDM'17 - Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, Cambridge, UK, Feb. 2017, pp. 781–789.

[43] Y. Xie, Q. Liu, Y. Zhou, B. Liu, Z. Wang, and J. Peng, "Off-policy evaluation and learning from logged bandit feedback: Error reduction via surrogate policy," in *7th International Conference on Learning Representations, ICLR 2019*, New Orleans, LA, USA, May 2019.

[44] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 4, p. 70–73, Apr. 2014.

[45] H. Dette and J. Gösmann, "A likelihood ratio approach to sequential change point detection for a general class of parameters," *Journal of the American Statistical Association*, Jul. 2019, Latest Articles.

[46] T. R. Karl and C. N. Williams, "An approach to adjusting climatological time series for discontinuous inhomogeneities," *Journal of Climate and Applied Meteorology*, vol. 26, no. 12, pp. 1744–1763, Dec. 1987.

[47] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 663–685, 1952.

[48] Y.-X. Wang, A. Agarwal, and M. Dudík, "Optimal and adaptive off-policy evaluation in contextual bandits," in *The 34th International Conference on Machine Learning (ICML 2017)*, vol. 70, Aug. 2017, pp. 3589–3597.

[49] L. Li, W. Chu, J. Langford, and X. Wang, "Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms," in *WSDM '11 - Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, Hong Kong, China, Feb. 2011, p. 297–306.

[50] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "Mobile edge computing partial offloading techniques for mobile urban scenarios," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018.

[51] A. Bozorgchenani, F. Mashhadi, D. Tarchi, and S. Salinas Monroy, "Multi-objective computation sharing in energy and delay constrained mobile edge computing environments," *IEEE Trans. Mobile Comput.*, 2020, Early Access.

[52] *Characteristics of terrestrial IMT-Advanced systems for frequency sharing/interference analyses*, ITU-R Report M.2292-0, Dec. 2013.

[53] *Further advancements for E-UTRA physical layer aspects*, 3GPP TR 36.814, Rev. 9.0.0, Mar. 2010.

[54] R. Agrawal, "Sample mean based index policies with O(log n) regret for the multi-armed bandit problem," *Advances in Applied Probability*, vol. 27, no. 4, pp. 1054–1078, Dec. 1995.

[55] Z. Zhu, T. Liu, S. Jin, and X. Luo, "Learn and pick right nodes to offload," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

Arash Bozorgchenani [S'17, M'20] is a Research Associate at Lancaster University, the UK. He has received a B.Sc. degree in Information Technology in 2013, and a M.Sc degree in Information Technology with a major in Computer Networks in 2016 in Iran. Later he has received a Ph.D. degree in Electronics, Telecommunications and Information Technology in 2020 at University of Bologna, Italy, where he also spent one year as a post-doctoral researcher. He has spent a 6-month research activity at university of Manitoba, Canada as a visiting PhD researcher. He has also been involved in both national Gaucho (PRIN 2015) and European (H2020 SANCUS) projects. His research interests are in the areas of resource allocation, machine learning and optimization techniques in wireless communications.

Setareh Maghsudi is an assistant professor at the University of Tübingen, and a senior research scientist at the Fraunhofer Heinrich-Hertz-Institute. During 2017-2020, she was an assistant professor at the Technical University of Berlin, where she also received her Ph.D. degree (summa cum laude) in 2015. During 2015-2017, she held post-doctoral positions at the University of Manitoba-Canada, and Yale University-USA. She has received several competitive fellowships, awards, and research grants, including from the German Research Foundation (DFG), German Ministry of Education and Research (BMBF), and Japan Society for the Promotion of Science (JSPS). Her main research interests lie in the intersection of network analysis and optimization, game theory, machine learning, and data science.

Daniele Tarchi [S'99, M'05, SM'12] was born in Florence, Italy, on 4th October 1975. He received his M.Sc. degree in Telecommunications Engineering and Ph.D. degree in Informatics and Telecommunications Engineering from the University of Florence, Florence, Italy, in 2000 and 2004, respectively.

From 2004 to 2010, he was a Research Associate with University of Florence, Italy. From 2010 to 2019 he was an Assistant professor at the University of Bologna, Bologna, Italy. He is now an Associate Professor at the University of Bologna, Italy. He is author of more than 100 papers, among which more than 30 on international journals. His research interests are mainly on resource allocation techniques, heterogeneous networks, Cognitive Radios and Networks, and Wireless Multimedia Networks, applied to both terrestrial and satellite scenarios. Recently he has mainly focused on Smart City scenarios, Multimedia systems, Fog Networks, and Edge Computing. He has been involved in several national and international research projects, and collaborates with several foreign research institutes.

Prof. Daniele Tarchi is an IEEE Senior Member since 2012. He is Editorial Board member for IEEE Wireless Communications Letters, IEEE Transactions on Vehicular Technology and IET Communications. He has been symposium co-chair for IEEE WCNC 2011, IEEE Globecom 2014, IEEE Globecom 2018 and IEEE ICC 2020, and workshop co-chair at IEEE ICC 2015.

Ekram Hossain (F'15) a Professor and Associate Head (Graduate Studies) in the Department of Electrical and Computer Engineering at University of Manitoba, Canada. He is a Member (Class of 2016) of the College of the Royal Society of Canada, a Fellow of the Canadian Academy of Engineering, and a Fellow of the Engineering Institute of Canada. Dr. Hossain's current research interests include design, analysis, and optimization of wireless, mobile, cognitive, and green communication networks with emphasis on beyond 5G cellular networks. To date, his research works have received 33,000+ citations (in Google Scholar, with h-index = 94). He was elevated to an IEEE Fellow "for contributions to spectrum management and resource allocation in cognitive and cellular radio networks".

He received the 2017 IEEE ComSoc TCGCC (Technical Committee on Green Communications Computing) Distinguished Technical Achievement Recognition Award "for outstanding technical leadership and achievement in green wireless communications and networking". He was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2017, 2018, 2019, and 2020. Currently he serves as the Editor-in-Chief of IEEE Press. Previously he served as the Editor-in-Chief for the IEEE Communications Surveys and Tutorials (2012–2016).