

# Toward Large-Scale Hybrid Edge Server Provision: An Online Mean Field Learning Approach

Zhiyuan Wang<sup>1</sup>, Jiancheng Ye<sup>2</sup>, *Member, IEEE*, and John C. S. Lui<sup>3</sup>, *Fellow, IEEE*

**Abstract**—The efficiency of a large-scale edge computing system primarily depends on three aspects: i) edge server provision, ii) task migration, and iii) computing resource configuration. In this paper, we study the dynamic resource configuration for hybrid edge server provision under two decentralized task migration schemes. We formulate the dynamic resource configuration as an online cost minimization problem, aiming to jointly minimize performance degradation and operation expenditure. Due to the stochastic nature, it is an online learning problem with partial feedback. To address it, we derive a deterministic mean field model to approximate the stochastic edge computing system. We show that the mean field model provides the increasingly accurate full feedback as the system scales. We then propose a learning policy based on the mean field model, and show that our proposed policy performs asymptotically as well as the offline optimal configuration. We provide two ways of setting the policy parameters, which achieve a constant competitive ratio (under certain mild conditions) and a sub-linear regret, respectively. Numerical results show that the mean field model significantly improves the convergence speed. Moreover, our proposed policy under the decentralized task migration schemes considerably reduces the operating cost (by 23%) and incurs little communication overhead.

**Index Terms**—Edge computing, online learning, mean field theory.

## I. INTRODUCTION

### A. Background and Motivation

THE recent proliferation of smart city and Internet-of-Things (IoT) applications has been driving a rapid growth of connected devices (e.g., IoT sensors and mobile users) [2]. These devices are the *sources* that repeatedly generate computing tasks of various delay-sensitive services. Edge computing,

providing computation resource in close proximity to the sources, is a promising paradigm to reduce the latency for many network applications [3]. Moreover, the exponential growth of artificial intelligent applications creates an urgent need to unleash the benefit of the edge network [4]. The performance of the large-scale edge computing system depends primarily on three factors: 1) edge server provision, 2) task migration scheme, and 3) dynamic resource configuration, which are the main focus of our study in this paper.

1) *Edge Server Provision*: The edge computing resource is typically the edge servers in close proximity to the sources. In practice, the edge server could be a micro datacenter or a server attached to an access point [5]. In general, the edge servers may function in the passive mode or the active mode:

- The edge server in the *passive mode* will admit and process the tasks offloaded by the sources (e.g., IoT sensors). The interaction between the sources and the passive-mode edge servers corresponds to the previous studies on computation task offloading (e.g., [6]–[11]).
- The edge server in the *active mode* will not directly admit the tasks offloaded by the sources, but will assist the passive-mode edge servers to process the waiting tasks. The interaction between the active-mode and passive-mode edge servers corresponds to the previous studies on edge collaboration (e.g., [12]–[17]).

We provide an illustrative edge network in Fig. 1, which consists of  $M = 10$  sources,  $N = 5$  passive-mode edge servers, and  $K = 1$  active-mode edge server. The ratios  $\theta \triangleq N/M$  and  $\eta \triangleq K/M$  represent the network operator's hybrid edge server provision. Given the above two modes, a source's computing task will first reach one of the passive-mode edge servers, and may be extracted by an idle active-mode edge server later. Therefore, the task execution progress is closely related to the adopted task migration scheme, which is discussed in the following.

2) *Task Migration Scheme*: The task migration scheme should adapt to the aforementioned edge server provision. An appropriate task migration scheme can significantly reduce the task execution latency, thus improves the Quality of Experience (QoE) perceived by the sources. There have been many studies on task migration under the coordination of the network operator who has the global information (e.g., [14], [17]). In real-world applications, however, it is costly to keep track of all the required information globally and persistently, especially when the network scale (e.g., number of sources and edge servers) is large. This motivates us to consider the more practical migration scheme for the hybrid

Manuscript received 14 December 2021; revised 12 March 2022; accepted 23 April 2022. Date of publication 8 June 2022; date of current version 18 July 2022. The work of Zhiyuan Wang was supported in part by the National Key Research and Development Program (NKRDP) of China under Grant 2019YFB1802803 and in part by The Chinese University of Hong Kong (CUHK) under Grant 6905407. The work of John C. S. Lui was supported in part by the Research Grants Council (RGC) under Grant SRF52122-4S02. An earlier version of this paper was presented in part at ACM MobiHoc 2021 [1] [DOI: 10.1145/3466772.3467042]. (*Corresponding author: Jiancheng Ye.*)

Zhiyuan Wang is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: zhiyuanwang@buaa.edu.cn).

Jiancheng Ye is with the Network Technology Laboratory and the Hong Kong Research Center, Huawei Technologies Company Ltd., Hong Kong (e-mail: yejiancheng@huawei.com).

John C. S. Lui is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: csui@cse.cuhk.edu.hk).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JSAC.2022.3180781>.

Digital Object Identifier 10.1109/JSAC.2022.3180781



Fig. 1. An illustrative scenario with  $M = 10$  sources,  $N = 5$  passive-mode edge servers, and  $K = 1$  active-mode edge server. The edge servers can be interconnected via a metropolitan-area-network or local-area-network.

edge server provision. In this paper, we will focus on two load-balancing policies, i.e., Join-Shortest-Queue (JSQ) and Longest-Queue-First (LQF). As shown in Fig. 1, the two migration schemes work as follows:

- JSQ( $d$ ) with  $d \in \{1, 2, \dots, N\}$ : Upon generating a task, the source probes  $d$  passive-mode edge servers uniformly at random, and migrates the task to the least loaded one among the  $d$  samplings.
- LQF( $b$ ) with  $b \in \{1, 2, \dots, N\}$ : Whenever an active-mode edge server has any capacity, it probes  $b$  passive-mode edge servers uniformly at random, and extracts a waiting task from the one with the heaviest workload among the  $b$  samplings.

Note that both JSQ( $d$ ) and LQF( $b$ ) aim to balance the workload among the passive-mode edge servers, thus improve the resource utilization and expedite the task execution. Furthermore, as the parameters  $d$  and  $b$  increase, JSQ( $d$ ) and LQF( $b$ ) are approaching to the classic join-the-shortest-queue and serve-the-longest-queue disciplines, respectively. As one can imagine, probing the edge servers results in communication overhead, which increases linearly in the two parameters  $d$  and  $b$ . The previous studies (e.g., [18], [19]) have shown that a small value can already ensure a good performance in the heavy-demand scenario. However, it is not clear yet whether this still holds when the operator needs to configure the available computing resource in a dynamic fashion (i.e., the third focus in this paper).

3) *Resource Configuration*: Despite the extensive studies on the task migration, the computing resource allocation has been overlooked in edge computing. A fixed resource configuration will inevitably result in either a low resource utilization or a poor system performance due to the workload fluctuation from the sources. Therefore, it is crucial to configure the computing resource in a dynamic fashion for the edge servers. Note that the “dynamic configuration” naturally relies on the underlying computation demand and the expenditure of running the resource. Both of them are priori unknowns and possibly time-varying in practice. This means that it is imperative to study the “online resource configuration” for the edge servers. The above discussions lead to the following key questions in this paper:

*Question 1: How to optimize the resource configuration under the hybrid edge servers provision in a dynamic fashion?*

*Question 2: Can one harness the benefits from JSQ( $d$ ) and LQF( $b$ ) even when  $d$  and  $b$  are small in the dynamic scenario?*

The major challenge of resolving the two key questions is the partial feedback issue resulted from the large-scale edge network. To overcome this challenge, we will introduce a mean field model to estimate the large-scale edge network,

and propose an *online mean field aided configuration policy*. We believe the results in this paper could lay the groundwork for using the mean field theory to analyze and optimize the dynamic hybrid edge server provision for the large-scale edge computing network.

## B. Main Results and Key Contributions

In this paper, we study the multi-period operation of the large-scale edge computing system with hybrid edge server provision, and we aim to minimize the total operating cost in an online fashion. At the beginning of each period (e.g., every hour), the network operator determines the resource configuration for the edge servers of two modes. During this period, the passive-mode edge servers admit computation tasks from the sources under the migration policy JSQ( $d$ ). The active-mode edge servers assist the passive-mode ones in executing the waiting tasks according to the migration policy LQF( $b$ ). At the end of each period, the network operator observes the cost of operating the network within this period, and then determines the resource configuration for the next period. Due to the stochastic nature of the large-scale edge network, it is difficult to anticipate the cost of other configuration decisions that were not adopted. Therefore, the dynamic resource configuration problem naturally exhibits the *partial feedback* issue. The main results and key contributions in this paper are as follows:

- *Problem Formulation*: We investigate the dynamic resource configuration for hybrid edge server provision under two decentralized task migration schemes. The goal is to jointly minimize the operation expenditure and the performance degradation in an online fashion. To the best of our knowledge, this is the first study on dynamic resource configuration for edge computing.
- *Resolving Partial Feedback via Mean Field*: To address the partial feedback, we introduce a deterministic mean field model to represent the large-scale stochastic edge network. We show that the original stochastic system converges to the deterministic mean field model as the system size and the period duration increase. As far as we know, we are the first to integrate the mean field theory and online learning with the partial feedback issue.
- *Online Mean Field aided Configuration Policy*: We devise an online mean field aided configuration policy for the large-scale edge computing system. Our proposed policy first discretizes the metric space, and then continuously explores and exploits the finite configuration candidates based on the mean field cost. We show that the cost incurred by our proposed policy is less than the sum between a constant multiple of the offline minimal cost and an extra constant. We provide two ways of setting the policy parameters, which achieve the constant competitive ratio (under certain mild condition) and the sub-linear regret, respectively.
- *Performance Evaluation*: We carry out extensive evaluation using the real-world electricity market data. The results show that the mean field model can significantly improve the convergence speed in the online resource configuration. Moreover, our proposed mean field aided

TABLE I  
RELATED LITERATURES ON EDGE COMPUTING

Literature	Task Migration		Resource Configuration	Dynamic Model
	Edge-Edge	Edge-Cloud		
[6]–[11]	×	✓	×	✓
[12][13]	✓	×	×	×
[14]–[17]	✓	✓	×	×
[20]–[23]	×	✓	✓	×
<b>This Paper</b>	✓	✓	✓	✓

configuration policy under the migration schemes JSQ(2) and LQF(2), considerably reduces the total operating cost (by 23%) and incurs little communication overhead.

The remainder of this paper is as follows. Section II reviews related literature. Section III introduces the system model and the problem formulation. Section IV derives a mean field model. Section V proposes the online resource configuration policy. Section VI provides the numerical results. We conclude this paper in Section VII.

## II. LITERATURE REVIEW

We review three streams of studies related to this paper, including edge computing, load balancing in multi-server system, and online learning algorithms.

### A. Edge Computing

There have been many excellent studies on edge computing [5]. We will focus on the recent studies that are mostly related to ours. Table I summarizes the related literature in terms of task migration, resource configuration, and other modeling features.

1) *Edge Server Provision & Task Migration*: The passive-mode edge server provision in this paper is related to the previous studies on the task offloading problems. Chen *et al.* in [6] study this problem as a potential game. The later studies further take into account the energy-efficiency aspect (e.g., [7]) and the service caching (e.g., [8], [9]). Ng *et al.* in [10] propose a two-phase stochastic coded offloading scheme, which minimizes the cost of the network, energy consumption by the UAVs. Asheralieva *et al.* in [11] further take into account privacy-preserving design for coded offloading. The active-mode edge server provision is related to edge collaboration. Sahni *et al.* in [12] study how to jointly schedule the tasks and the network flows in the collaborative edge computing. Galanopoulos *et al.* in [13] consider the cooperative IoT data analytics on the edge nodes. More recently, Tang *et al.* in [14] propose a general 3C resource sharing framework, which takes into account communication, computation, and caching. Ndikumana *et al.* in [15] propose an edge server collaboration paradigm, and jointly consider the computing, caching, communication, and control. Peng *et al.* in [16] take the leasing cost into consideration, and design intelligently joint caching and offloading strategies. Moreover, Poularakis *et al.* in [17] formulate a static service placement and request routing problem, which can generalize several

previous studies on edge computing. They also propose an algorithm that achieves the close-to-optimal performance.

2) *Resource Configuration*: The resource configuration problem has not been widely studied in edge computing before. A few studies consider a static scenario. Zhang *et al.* in [20] investigate how to allocate the computing resource of edge and cloud servers, and propose a distributed optimization framework. Kiani and Ansari [21] consider a hierarchical model to allocate the computing resource based on an auction-based method. Ng *et al.* in [22] propose a double auction mechanism to allocate the resource of the edge servers under the PolyDot codes. Chen *et al.* in [23] focus on the energy consumption and study the optimal allocation of both computation and communication resource. Meskar and Liang [24] focus on the fairness issue of multi-resource allocation for edge servers. Furthermore, the dynamic configuration in edge computing is seldom studied. Zhou *et al.* in [25] investigate the dynamic server provision in IoT data streaming, but do not address the resource allocation problem.

Our study in this paper differs from the above works on edge computing in the following three aspects.

- First, we resolve the dynamic resource configuration problem in the online setting, taking into account the computing demand uncertainty and the unknown operation expenditure. This is a critical issue for edge computing deployment, but is seldom studied before.
- Second, we take into account the hybrid edge server provision together with two decentralized task migration schemes. Such a paradigm can generalize several previous studies into a unified framework.
- Third, we investigate the limiting system performance when the number of sources is large via the mean field model. This is a practically important issue for the large-scale edge network deployment.

Overall, the three aspects are mutually affected, which renders the online configuration policy design highly non-trivial.

### B. Load Balancing in Multi-Server System

The task migration schemes in this paper are related to the load-balancing policies in multi-server systems, which study how the dispatcher routes the incoming jobs to different servers [26]. In this problem, mean field approximation has been widely used to investigate the steady state of the limiting system. Mitzenmacher in [18] characterizes the average response time under JSQ(2) for the M/M/1 system, which unveils an exponential improvement compared to the random dispatching. Later works extend this study from the perspective of batch-job arrival (e.g., [19]), general serving time distribution (e.g., [27]), resource budget (e.g., [28]), and resource pooling (e.g., [29]). The above studies only show that the original system converges to the mean field model as the system size increases, but do not analyze the approximation error. More recently, Ying in [30] explicitly derives the approximation error in terms of the system size.

In this paper, the task migration scheme from the sources to the passive-mode edge servers shares a similar idea with the above studies (e.g., [18]). Moreover, the migration scheme for

the active-mode edge servers is a generalization of the resource pooling study in [29].

### C. Online Learning Algorithms

The dynamic resource configuration in this paper corresponds to the online learning problem with partial feedback on the metric space. To resolve this problem, this paper proposes an online mean field aided policy, which consists of the discretization and learning phases. Specifically, the mean field model provides the approximated full feedback. Moreover, the learning phase is primarily based on the multiplicative weight update (MWU) method [31], which originates from the classic problem “prediction with expert advices” [32]. However, the presence of discretization phase renders the regret analysis of the proposed algorithm substantially different from the standard procedure of the classic MWU method.

## III. SYSTEM MODEL

We consider a set  $\mathcal{M} = \{1, 2, \dots, M\}$  of  $M$  sources (e.g., IoT sensors or mobile users), which repeatedly generate delay-sensitive computing tasks. The edge computing operator adopts a hybrid edge server provision in proximity to the sources. Specifically, there are a set  $\mathcal{N} = \{1, 2, \dots, N\}$  of  $N$  passive-mode edge servers and a set  $\mathcal{K} = \{1, 2, \dots, K\}$  of  $K$  active-mode edge servers. The task migration scheme under the hybrid edge server provision is as follows:

- The sources can offload their computing tasks to one of the  $N$  passive-mode edge servers, but cannot directly access the active-mode edge servers on their own.
- The  $K$  active-mode edge servers will assist the  $N$  passive-mode edge servers to process the waiting tasks by extracting a waiting task.

For notation simplicity, we define  $\theta \triangleq N/M$  and  $\eta \triangleq K/M$ . Accordingly, the tuple  $(\theta, \eta)$  represents the operator’s hybrid edge server provision, which reflects the operator’s long-term infrastructure deployment. For example, if the operator decides to deploy 10 passive-mode edge servers and 1 active-mode edge server for every 100 IoT sensors within the region of its interest, then we have  $(\theta, \eta) = (0.1, 0.01)$ . Later on, we will investigate how the edge computing system scales as  $M$  increases. This captures the rapid growth of the delay-sensitive applications in the future large-scale network.

Furthermore, the task generation rate of each source  $m \in \mathcal{M}$  could be time-varying and unpredictable in practice. Therefore, given the hybrid edge server provision  $(\theta, \eta)$ , the operator will dynamically configure the available computing resource of the  $N + K$  edge servers. We consider an operation horizon with a set  $\mathcal{T} = \{1, 2, \dots, T\}$  of periods (e.g., 1000 hours). Each period  $t \in \mathcal{T}$  has the equal time duration  $\delta$  (e.g., 1 hour), and we let  $\tau \in [0, \delta]$  be the time index within each period  $t$ . The operator configures the available resource of each edge server at the beginning of each period  $t$ , and then the edge computing system runs under this configuration until the end of this period.

We introduce the network model and characterize the network state in Section III-A and Section III-B, respectively.

TABLE II  
KEY NOTATIONS

Symbol	Physical Meaning
$\mathcal{M}$	A set $\mathcal{N} = \{1, 2, \dots, N\}$ of sources.
$\mathcal{N}$	A set $\mathcal{N} = \{1, 2, \dots, N\}$ of passive-mode edge servers.
$\mathcal{K}$	A set $\mathcal{K} = \{1, 2, \dots, K\}$ of active-mode edge servers.
$(\theta, \eta)$	The hybrid edge server provision mode
$d$	The migration parameter of passive-mode edge servers.
$b$	The migration parameter of active-mode edge servers.
$\lambda_t^m$	Task generating rate of source $m \in \mathcal{M}$ in period $t$ .
$\phi_{t[i]}^m$	The $i$ -th task generated by source $m \in \mathcal{M}$ in period $t$ .
$\Phi_t$	The tasks generated in period $t$ .
$x_t$	Resource of a passive-mode edge server in period $t$ .
$y_t$	Resource of an active-mode edge server in period $t$ .
$z_t$	Resource configuration decision in period $t$ .
$Q_t^n(\tau)$	Number of waiting tasks at server $n$ at time $\tau \in [0, \delta]$ .
$L_t^{[N]}(\cdot)$	Average workload, defined in Eq. (3).
$L_t^{[N \delta]}(\cdot)$	Time-average workload, defined in Eq. (4).
$C_t^{[N \delta]}(\cdot)$	The Operating cost of period $t$ , defined in Eq. (9).
$S_i^{[N]}(\tau)$	Density-based state defined in Eq. (11).
$\tilde{s}$	The fixed point of the mean field model.
$l(\cdot)$	Mean field time-average workload defined in Eq. (21).
$C_t(\cdot)$	Mean field cost defined in Eq. (25).
$\mathfrak{A}$	The proposed online mean field aided policy.
$(\beta, \epsilon)$	The parameters in policy $\mathfrak{A}$
$C_T^{\mathfrak{A}}$	Total mean field cost under policy $\mathfrak{A}$ across $T$ periods
$C_T^*$	Total mean field cost under offline optimal configuration.

We then formulate the operator’s dynamic resource configuration problem in Section III-C. Table II summarizes the key notations.

### A. Network Model

We characterize the edge computing network based on the sources’ computation tasks, the migration schemes, and the computing resource.

1) *Computation Task*: In each period  $t$ , source  $m \in \mathcal{M}$  will generate multiple computation tasks. We model the stochastic nature of the task generation based on the generating time and the computing intensity.

a) *Generating time*: We follow the previous studies (e.g., [8]) and model the task generation of each source  $m \in \mathcal{M}$  as the Poisson process with the rate  $\lambda_t^m$  in period  $t$ . Note that the rate  $\lambda_t^m$  may vary over different periods, which captures the demand fluctuation of source  $m$ . Accordingly, we let  $\tau_{t[i]}^m \in [0, \delta]$  denote the generating time of the  $i$ -th task of source  $m$  in period  $t$ . To facilitate our later discussion, we denote the average task generating rate in period  $t$  as follows

$$\bar{\lambda}_t^{[M]} \triangleq \sum_{m=1}^M \frac{\lambda_t^m}{M}, \quad \forall t \in \mathcal{T}, \quad (1)$$

where the superscript represents that the average is taken over the  $M$  sources.

b) *Computing Intensity*: The computing intensity of a task represents its complexity and can be roughly measured by the required CPU cycles [6]. Mathematically, we let  $l_{t[i]}^m$

denote the computing intensity of the  $i$ -th task of source  $m \in \mathcal{M}$  in period  $t$ . We follow the previous empirical studies (e.g., [33], [34]) and model  $l_{t[i]}^m$  as an exponentially distributed random variable with a normalized mean value. Our later analysis can be extended to the general distributions, which will be elaborated at the end of Section IV.

The tuple  $\phi_{t[i]}^m \triangleq \{\tau_{t[i]}^m, l_{t[i]}^m\}$  represents the  $i$ -th computation task of source  $m \in \mathcal{M}$  in period  $t$ . We let  $\phi_t^m \triangleq \{\phi_{t[1]}^m, \phi_{t[2]}^m, \phi_{t[3]}^m, \dots\}$  denote the task profile of source  $m$  in period  $t$ . Accordingly,  $\Phi_t \triangleq (\phi_t^m : \forall m \in \mathcal{M})$  represents the task profile of the entire system in period  $t$ .

2) *Passive-Mode Edge Server Provision*: Each source  $m \in \mathcal{M}$  may offload its tasks  $\phi_t^m$  to one of the  $N$  passive-mode edge servers. Different from the previous studies on task offloading (e.g., [6]–[9]), we will focus on a decentralized migration scheme JSQ( $d$ ), where  $d \in \{1, 2, \dots, N\}$ . It works in two steps as follows:

- Upon generating a task (e.g.,  $\phi_{t[i]}^m$  at time  $\tau_{t[i]}^m \in [0, \delta]$ ), the source  $m \in \mathcal{M}$  inquires about the number of tasks in  $d$  passive-mode edge servers, which are uniformly selected at random.
- The source  $m$  migrates the task  $\phi_{t[i]}^m$  to the one holding the least tasks among the  $d$  samplings.

Note that the above procedure tends to balance the workload among the  $N$  passive-mode edge servers. A larger parameter  $d$  would lead to a better performance, but also increases the communication overhead. Previous studies (e.g., [18], [19]) have shown that a small parameter (e.g.,  $d = 2$ ) can already ensure a good performance in the heavy-demand scenario. We will further investigate the impact of the migration parameter  $d$  in Section VI-B.

3) *Active-Mode Edge Server Provision*: The  $K$  active-mode edge servers will assist the passive-mode edge servers to execute the waiting tasks offloaded by the sources. Different from the previous studies on edge collaboration (e.g., [12]–[15], [17]), we focus on a decentralized scheme LQF( $b$ ), where  $b \in \{1, 2, \dots, N\}$ . It works in two steps as follows:

- Upon being idle, the active-mode edge server  $k \in \mathcal{K}$  selects  $b$  passive-mode edge servers uniformly at random, and inquires about the number of tasks at the  $b$  selected passive-mode edge servers.
- The active-mode edge server  $k$  will extract a waiting task from the most loaded one among the  $b$  samplings according to the FIFO rule.

Similarly, the procedure in LQF( $b$ ) tends to balance the workload among the  $N$  passive-mode edge servers. A larger  $b$  leads to a better performance in terms of balancing the workload, but also increases the communication overhead. Section VI will show that a small parameter (e.g.,  $b = 2$ ) can already ensure a good performance.

So far, we have elaborated the task migration scheme. Next we move on to model the computing resource.

4) *Computing Resource*: Each edge server is equipped with a certain amount of computing resource. The operator needs to configure the available computing resource (e.g., the number of VMs) at the beginning of each period  $t$  without

the knowledge of the upcoming tasks  $\Phi_t$ . Mathematically, we let  $x_t$  denotes the CPU frequency (in cycles per second) of each passive-mode edge server. Hence the total available computing resource at the passive-mode edge servers is  $Nx_t$ . Similarly, we let  $y_t$  denote the computing resource at each active-mode edge server, thus the total computing resource at the active-mode edge server is  $Ky_t$ . The tuple  $z_t = (x_t, y_t)$  denotes the resource configuration in period  $t$ . The operator chooses  $z_t$  in the metric space  $\mathcal{Z}$ , which is defined as

$$\mathcal{Z} \triangleq \{(x, y) \mid x_L \leq x \leq x_H, y_L \leq y \leq y_H\}, \quad (2)$$

where the feasible ranges  $[x_L, x_H]$  and  $[y_L, y_H]$  depend on the hardware setup in practice. Recall that the operator has no task information until the period end. That is, the operator would decide  $z_t \in \mathcal{Z}$  without relying on  $\Phi_t$  and  $\lambda_t$ .

## B. Network Characterization

We characterize the network state and introduce the performance metric based on the above network model.

1) *Network State*: In each period  $t$ , we let  $Q_t^n(\tau) \in \mathcal{B}$  denote the number of tasks in the passive-mode edge server  $n \in \mathcal{N}$  at time  $\tau \in [0, \delta]$ , where  $\mathcal{B} \triangleq \{0, 1, \dots, B\}$  and  $B$  is the buffer size. Accordingly,  $\mathbf{Q}_t(\tau) = \{Q_t^n(\tau) \in \mathcal{B} : \forall n \in \mathcal{N}\}$  represents the network state at time  $\tau$  in period  $t$ . Note that  $\{\mathbf{Q}_t(\tau) \in \mathcal{B}^N : \forall \tau \in [0, \delta]\}$  is an  $N$ -dimensional continuous time Markov chain (CTMC). We have two-fold elaboration on the dependency of the network state:

- Given the migration scheme JSQ( $d$ ),  $Q_t^n(\tau)$  depends on the task profile of the passive-mode edge server  $n$ , as well as the task profiles of other passive-mode edge servers.
- Given the migration scheme LQF( $b$ ),  $Q_t^n(\tau)$  depends on the resource configuration at both the passive-mode and active-mode edge servers.

To emphasize the above dependencies, we will sometimes use  $Q_t^n(\tau, z_t, \Phi_t)$  to denote the number of tasks in the passive-mode edge server  $n$  at time  $\tau$ . Accordingly, the *average workload* among the  $N$  passive-mode edge servers at time  $\tau \in [0, \delta]$  is given by

$$L_t^{[N]}(\tau, z_t, \Phi_t) \triangleq \frac{1}{N} \sum_{n=1}^N Q_t^n(\tau, z_t, \Phi_t), \quad (3)$$

where the superscript  $[N]$  represents that the average is taken over the  $N$  passive-mode edge servers. Moreover, the *time-average workload* in period  $t$  is given by

$$L_t^{[N][\delta]}(z_t, \Phi_t) \triangleq \frac{1}{\delta} \int_0^\delta L_t^{[N]}(\tau, z_t, \Phi_t) d\tau, \quad (4)$$

where the superscript  $[\delta]$  represents that the average is taken over the period duration  $\delta$ . Next we elaborate why the time-average workload Eq. (4) is a crucial performance metric for the operator to optimize.

2) *Performance Metric*: The operator aims to expedite the task execution and improve the Quality of Experience (QoE) of the  $M$  sources. Hence the execution latency (i.e., the time that a task spends in the system) is a crucial performance metric for the operator to optimize. Specifically, Eq. (4) implies that

the time-average number of tasks in all the passive-mode edge servers is  $NL_t^{[N]|\delta]}(z_t, \Phi_t)$ . The sources' total task generating rate is  $M\bar{\lambda}_t^{[M]}$ . By Little's Law [35], the average execution latency of the tasks  $\Phi_t$  in period  $t$  is

$$\frac{N \cdot L_t^{[N]|\delta]}(z_t, \Phi_t)}{M \cdot \bar{\lambda}_t^{[M]}} = \frac{\theta \cdot L_t^{[N]|\delta]}(z_t, \Phi_t)}{\bar{\lambda}_t^{[M]}}, \quad (5)$$

where  $\theta = N/M$  is the passive-mode edge server provision ratio. Therefore, the time-average workload in Eq. (4) is proportional to the task execution latency. In Section III-C, we will model the network performance degradation based on it.

### C. Problem Formulation

We first define the operating cost, and then introduce the problem formulation of the dynamic resource configuration.

1) *Operating Cost*: We model the operating cost based on the operation expenditure and the performance degradation.

a) *Operation expenditure*: is the monetary cost of the energy consumption for the operator. Such a cost has a positive relation with the resource configuration decision  $z_t = (x_t, y_t)$ . We will use a general polynomial convex structure to model the operator's monetary cost. To be specific, for each passive-mode edge server, we model the monetary cost under the resource configuration  $x_t$  as follows

$$\xi_t^P x_t^\sigma, \quad (6)$$

where  $\sigma \geq 1$  and  $\xi_t^P \in [0, \xi_{max}^P]$ . Similarly, given the computing resource configuration  $y_t$ , we model the cost incurred by each active-mode edge server as

$$\xi_t^A y_t^\sigma, \quad (7)$$

where  $\xi_t^A \in [0, \xi_{max}^A]$ . We have two-fold elaboration regarding the above cost structure.

- In the real-world network, the two coefficients  $\xi_t^P$  and  $\xi_t^A$  depend on many unpredictable factors such as the infrastructure management, the energy consumption, and the electricity price. Hence the operator does not know ( $\xi_t^P, \xi_t^A$ ) until the end of period  $t$ .
- Our later analysis is applicable to any parameter  $\sigma \geq 1$ , which corresponds a general convex cost structure in the polynomial form. Note that the quadratic structure (i.e.,  $\sigma = 2$ ) could capture the power scaling phenomenon (as in [36]–[38]). Moreover, the linear structure (i.e.,  $\sigma = 1$ ) could capture the case where each edge server consists of multiple VMs (as in [39]).

Based on the above monetary cost, the resource configuration  $z_t = (x_t, y_t)$  incurs the operation expenditure  $\xi_t^P N x_t^\sigma + \xi_t^A K y_t^\sigma$  in period  $t$ . As we will see later, it is practically important to investigate how the system scales when the number of sources  $M$  increases. To facilitate our later analysis, we will focus on the following *average operation expenditure* in each period  $t$

$$\frac{\xi_t^P N x_t^\sigma + \xi_t^A K y_t^\sigma}{M} = \xi_t^P \theta x_t^\sigma + \xi_t^A \eta y_t^\sigma, \quad (8)$$

where  $\theta = N/M$  and  $\eta = K/M$  represent the operator's hybrid edge server provision.

b) *Performance degradation*: measures the sources' QoE reduction due to the increase in latency. In each period  $t$ , we measure the QoE reduction based on the time-average workload  $L_t^{[N]|\delta]}(z_t, \Phi_t)$  defined in Eq. (4). Mathematically, we adopt a general formulation and let  $G(L)$  denote the degradation given the time-average workload  $L$ . Specifically,  $G(L)$  is continuous and increasing in  $L$  with  $G(0) = 0$ .

Based on the discussion above, we define the operator's cost in period  $t$  as follows:

$$C_t^{[N]|\delta]}(z_t, \Phi_t) \triangleq G\left(L_t^{[N]|\delta]}(z_t, \Phi_t)\right) + \xi_t^P \theta x_t^\sigma + \xi_t^A \eta y_t^\sigma, \quad (9)$$

which comprises the performance degradation and the average operation expenditure. The operator can flexibly choose the function  $G(\cdot)$  to balance how much it prioritizes the system performance over the monetary expenditure. That is, given an appropriate function  $G(\cdot)$ , the operator achieves its desired outcome by minimizing the cost in Eq. (9) over  $z_t \in \mathcal{Z}$ .

2) *Operator's Problem*: The operator determines the resource configuration  $z_t$  sequentially at the beginning of period  $t$ , aiming to minimize the total cost during the  $T$  periods. However, the operator cannot observe the task profile  $\Phi_t$  and the per-unit operation expenditure ( $\xi_t^P, \xi_t^A$ ) until the end of period  $t$ . That is, the operator needs to solve the following online cost minimization problem:

**Problem 1 (Online Cost Minimization Problem)**:

$$\begin{aligned} \min \quad & \sum_{t=1}^T C_t^{[N]|\delta]}(z_t, \Phi_t) \\ \text{s.t.,} \quad & z_t \in \mathcal{Z}, \quad \forall t \in \mathcal{T}. \end{aligned} \quad (10)$$

Problem 1 is an online optimization problem on the metric space. The key challenges to solve it are two-fold:

- **Partial Feedback**: Problem 1 exhibits the partial feedback issue in terms of the performance degradation. Specifically, the operator can observe the performance degradation after adopting the resource configuration  $z_t$ . However, it is hard for the operator to anticipate the performance of other configuration due to the stochastic nature of the large-scale edge network when there are a great number of sources.
- **Lack for Gradient Information**: The operator does not know the explicit gradient of the operating cost  $C_t^{[N]|\delta]}(z_t, \Phi_t)$  with respect to  $z_t$ , let alone its convexity or the Lipschitz condition. Therefore, the gradient-based learning algorithms (e.g., online gradient descent [40]) do not work in Problem 1. Moreover, we are not able to follow the learning algorithms (e.g., [41]–[43]) that rely on the Lipschitz constant.

To overcome the above challenges, we first introduce how to tackle the partial feedback issue via the *mean field theory* in Section IV. We then propose an *online mean field aided policy* to address the unknown information in Section V.

## IV. MEAN FIELD MODEL

In Section III, we characterize the hybrid edge network as an  $N$ -dimensional stochastic CTMC. In this section, we introduce a *deterministic* mean field model that approximates the

$N$ -dimensional stochastic CTMC. Mathematically, we want to estimate the time-average workload  $L_t^{[N]\delta}(\mathbf{z}, \Phi_t)$  for any configuration  $\mathbf{z} \in \mathcal{Z}$  based on the mean field model. For notation simplicity, we will focus on a generic period and suppress the period index  $t$  in this section.

We first introduce the density-based state and the mean field model in Section IV-A and Section IV-B, respectively. We then study the relationship between the deterministic mean field model and the original stochastic system in Section IV-C.

#### A. Density-Based State

Recall that  $Q^n(\tau)$  represents the number of tasks in passive-mode edge server  $n \in \mathcal{N}$  at time  $\tau \in [0, \delta]$ . That is,  $\mathbf{Q}(\tau) = \{Q^n(\tau) \in \mathcal{B} : \forall n \in \mathcal{N}\}$  is a *quantity-based* state characterization. Now we introduce a *density-based* state and let  $S_i^{[N]}(\tau)$  denote the “fraction” of passive-mode edge servers holding at least  $i$  tasks at time  $\tau$ , i.e.,

$$S_i^{[N]}(\tau) \triangleq \frac{1}{N} \sum_{n=1}^N \mathbb{I}(Q^n(\tau) \geq i), \quad \forall i \in \mathcal{B}, \quad (11)$$

where  $\mathbb{I}(\cdot)$  is the indicator function and the superscript  $[N]$  represents that there are  $N$  passive-mode edge servers. According to the definition Eq. (11), we have  $S_0^{[N]}(\tau) = 1$  for any  $\tau \in [0, \delta]$ . Moreover,  $\mathbf{S}^{[N]}(\tau) = \{S_i^{[N]}(\tau) : \forall i \in \mathcal{B}\}$  is the *density-based* state characterization for the system. Proposition 1 presents the relationship between  $\mathbf{Q}(\tau)$  and  $\mathbf{S}^{[N]}(\tau)$ . The proof follows directly from the definition in Eq. (11).

*Proposition 1: The quantity-based state  $\mathbf{Q}(\tau)$  and the density-based state  $\mathbf{S}^{[N]}(\tau)$  satisfy the following equality*

$$\frac{1}{N} \sum_{n=1}^N Q^n(\tau) = \sum_{i=1}^{\mathcal{B}} S_i^{[N]}(\tau), \quad (12)$$

which represents the average workload  $L^{[N]}(\tau, \mathbf{z}, \Phi)$ .

Proposition 1 essentially introduces two ways of calculating the average workload  $L^{[N]}(\tau, \mathbf{z}, \Phi)$ . As we will see later, it is more convenient to depict the system dynamics based on the density-based state  $\mathbf{S}^{[N]}(\tau)$ . Therefore, we will derive the mean field model based on the density-based state.

#### B. Derivation of Mean Field Model

To derive the mean field model, we first introduce the state transition and the expected drift. Afterwards, we will formally define the mean field model.

1) *State Transition:* The density-based state  $\mathbf{S}^{[N]}(\tau)$  in our problem is associated with two types of transitions, i.e., task admission transition and task execution transition. Next, we will introduce the two transitions based on the vector  $\mathbf{e}^a = \{e_i^a : \forall i \in \mathcal{B}\}$  defined as  $e_i^a \triangleq \mathbb{I}(i = a)/N$  for any  $i \in \mathcal{B}$ .

- **Task admission transition** occurs whenever a passive-mode edge server admits a new task from the sources under the migration policy JSQ( $d$ ). If the task admission transition occurs to a passive-mode edge server holding  $a - 1$  tasks at time  $\tau$ , then  $S_a^{[N]}(\tau)$  increases by  $1/N$ , while the other elements of  $\mathbf{S}^{[N]}(\tau)$

do not change. In this case, the density-based state becomes  $\mathbf{S}^{[N]}(\tau) + \mathbf{e}^a$ .

- **Task execution transition** occurs whenever a task in a passive-mode edge server is completed or migrated to one of the active-mode edge servers. If the task execution transition occurs at a passive-mode edge server holding  $a$  tasks at time  $\tau$ , then  $S_a^{[N]}(\tau)$  decreases by  $1/N$ , while the other elements of  $\mathbf{S}^{[N]}(\tau)$  do not change. In this case, the density-based state becomes  $\mathbf{S}^{[N]}(\tau) - \mathbf{e}^a$ .

The set  $\mathcal{E} \triangleq \{\pm \mathbf{e}^a : \forall a \in \mathcal{B}\}$  contains all the state transitions. Next we introduce the expected drift of the density-based state  $\mathbf{S}^{[N]}(\tau)$  based on the state transitions above.

2) *Expected Drift:* The expected drift of the density-based state  $\mathbf{S}^{[N]}(\tau)$  at time  $\tau$  is defined as

$$F_i(\mathbf{S}^{[N]}(\tau)) \triangleq \lim_{\Delta \rightarrow 0} \frac{\mathbb{E} [S_i^{[N]}(\tau + \Delta) - S_i^{[N]}(\tau)]}{\Delta}, \quad \forall i \in \mathcal{B}, \quad (13)$$

which measures the increasing rate of  $S_i^{[N]}(\tau)$  on average. Note that the expected drift  $\{F_i(\cdot) : \forall i \in \mathcal{B}\}$  depicts how the density-based state  $\mathbf{S}^{[N]}(\tau)$  behaves on average. We present the expression of  $\{F_i(\cdot) : \forall i \in \mathcal{B}\}$  in Proposition 2 and elaborate the rationale in the following proof sketch.

*Proposition 2: Given the resource configuration  $(x, y)$  and the average task arrival rate  $\bar{\lambda}^{[M]}$ , the expected drift at the density-based state  $\mathbf{s} = \{s_i \in [0, 1] : \forall i \in \mathcal{B}\}$  is given by  $F_0(\mathbf{s}) = 0$  and*

$$F_i(\mathbf{s}) = \frac{\bar{\lambda}^{[M]}}{\theta} (s_{i-1}^d - s_i^d) - x(s_i - s_{i+1}) - \frac{\eta}{\theta} \cdot y \left[ (1 - s_{i+1})^b - (1 - s_i)^b \right], \quad \forall i \geq 1, \quad (14)$$

where the tuple  $(\theta, \eta)$  represents the hybrid edge server provision and we let  $s_{\mathcal{B}+1} = 0$  for consistency. Moreover,  $d$  and  $b$  are the parameters in the task migration schemes JSQ( $d$ ) and LQF( $b$ ), respectively.

*Proof:* For case  $i = 0$ ,  $F_0(\mathbf{s}) = 0$  directly follows from the definition in Eq. (11). For case  $i \geq 1$ , to derive Eq. (14), we consider a time interval  $[\tau, \tau + \Delta]$  and compute the following term

$$\mathbb{E} [S_i^{[N]}(\tau + \Delta) - S_i^{[N]}(\tau)]. \quad (15)$$

Based on the previous discussion, the task admission transition  $\mathbf{e}^i$  and the task execution transition  $-\mathbf{e}^i$  lead to the increment  $1/N$  and the decrement  $-1/N$  for  $S_i^{[N]}(\tau)$ , respectively. To calculate Eq. (15), we consider the expected times that the transitions  $\pm \mathbf{e}^i$  occur during the interval  $[\tau, \tau + \Delta]$ . Next we introduce an event (i.e., SA $_i$ ) leading to the task admission transition  $\mathbf{e}^i$  and another two events (i.e., PE $_i$  and AE $_i$ ) leading to the task execution transition  $-\mathbf{e}^i$ .

- **Source-task-admission** event SA $_i$  means that a new task from the sources arrives at one of the passive-mode edge servers holding exactly  $i - 1$  tasks under the migration scheme JSQ( $d$ ). First, each source  $m \in \mathcal{M}$  generates new tasks at the rate  $\lambda^m$ , thus there are  $\Delta \sum_{m=1}^M \lambda^m$  arrival tasks during the interval  $[\tau, \tau + \Delta]$  on average. Second, under the migration scheme JSQ( $d$ ), a new

task is eventually migrated to a passive-mode edge server holding exactly  $i - 1$  tasks with the probability  $[S_{i-1}^{[N]}(\tau)]^d - [S_i^{[N]}(t)]^d$ . Hence event  $\text{SA}_i$  leads to the following increment for Eq. (15):

$$\frac{1}{N} \Delta M \bar{\lambda}^{[M]} \left( [S_{i-1}^{[N]}(t)]^d - [S_i^{[N]}(t)]^d \right). \quad (16)$$

- **Passive-execution** event  $\text{PE}_i$  means that one of the passive-mode edge servers holding  $i$  tasks completes a task. First, there are  $[S_i^{[N]}(t) - S_{i+1}^{[N]}(t)]N$  passive-mode edge servers holding  $i$  tasks. Second, the exponentially distributed computing intensity (i.e.,  $l_{t[i]}^m \sim \text{Exp}(1)$ ) and the CPU frequency  $x$  of the passive-mode edge server indicate that event  $\text{PE}_i$  occurs at rate  $x$ . That is, there are  $\Delta x$  tasks completed during the interval  $[\tau, \tau + \Delta]$  on average. Hence event  $\text{PE}_i$  leads to the following decrement for Eq. (15):

$$-\frac{1}{N} \Delta x [S_i^{[N]}(t) - S_{i+1}^{[N]}(t)] N. \quad (17)$$

- **Active-execution** event  $\text{AE}_i$  means that the active-mode edge server completes a task for a passive-mode edge server holding  $i$  tasks under the migration scheme LQF( $b$ ). First, the heaviest load among the  $b$  samplings is  $i$  with the probability  $[1 - S_{i+1}^{[N]}(t)]^b - [1 - S_i^{[N]}(t)]^b$ . Second, the exponentially distributed computing intensity (i.e.,  $l_{t[i]}^m \sim \text{Exp}(1)$ ) and the CPU frequency  $Ky$  of all the active-mode edge servers indicate that event  $\text{AE}_i$  occurs at the rate  $Ky$ . Hence event  $\text{AE}_i$  leads to the following decrement for Eq. (15)

$$-\frac{1}{N} \Delta Ky \left( [1 - S_{i+1}^{[N]}(t)]^b - [1 - S_i^{[N]}(t)]^b \right). \quad (18)$$

Eq. (15) equals to the summation of (16)~(18). Substituting it into Eq. (13) leads to the expression in Eq. (14).<sup>1</sup>  $\square$

3) *Mean Field Model*: Next we will formally define the mean field model. For notation clarity, we will use the lower-case notations  $\{s_i(\tau) : \forall i \in \mathcal{B}\}$  to denote the mean field.

*Definition 1*: The mean field model  $\mathbf{s}(\tau) = \{s_i(\tau) \in [0, 1] : \forall i \in \mathcal{B}\}$ , is defined by the following three types of conditions:

- *Initial condition*  $\mathbf{s}(0) = \mathbf{s}^0$ .
- *Boundary condition*  $s_0(\tau) = 1, \forall \tau \geq 0$ .
- *Drift condition*  $\frac{d\mathbf{s}(\tau)}{d\tau} = \mathbf{F}(\mathbf{s}(\tau))$  where the set of functions  $\mathbf{F}(\mathbf{s}) = \{F_i(\mathbf{s}) : \forall i \in \mathcal{B}\}$  are given in (14).

Note that the mean field model in Definition 1 is a set of ordinary differential equations, which are deterministic. Specifically, the initial condition specifies where the mean field starts to evolve. The boundary condition coincides with the definition of the density-based state in Eq. (11). The drift condition is the same as the expected drift of the original stochastic system  $\mathbf{S}^{[N]}(\tau)$ . Next we introduce the connection between the deterministic mean field model  $\mathbf{s}(\tau)$  and the original stochastic system  $\mathbf{S}^{[N]}(\tau)$ .

<sup>1</sup>The mean field model is not limited to the exponentially distributed computing intensity. We elaborate this in detail at the end of Section IV.

### C. Fixed Point and Convergence

We first present the fixed point of the mean field model. We then introduce the convergence relation between the deterministic mean field model and the original stochastic system.

1) *Fixed Point*: The fixed point of the mean field model is a state  $\tilde{\mathbf{s}}$ , at which the mean field model does not change anymore. That is, we have  $\mathbf{s}(\tau') = \tilde{\mathbf{s}}$  for any  $\tau' \geq \tau$  if  $\mathbf{s}(\tau) = \tilde{\mathbf{s}}$ . Theorem 1 presents the fixed point of the mean field model. The proof is given in our technical report [44].

*Theorem 1*: The fixed point  $\tilde{\mathbf{s}} = \{\tilde{s}_i : \forall i \in \mathcal{B}\}$  of the mean field model in Definition 1 is given by

$$\tilde{s}_i = \begin{cases} 1, & \text{if } i = 0, \\ g(\tilde{s}_{i+1}; \gamma), & \text{if } 1 \leq i < B, \\ g(0; \gamma), & \text{if } i = B, \end{cases} \quad (19)$$

where the function  $g(s; \gamma)$  is

$$g(s; \gamma) \triangleq \left[ \frac{\theta xs - \eta y(1-s)^b + \gamma}{\bar{\lambda}^{[M]}} \right]^{\frac{1}{d}}. \quad (20)$$

Moreover, the constant  $\gamma$  solves  $g^{(1+B)}(0; \gamma) = 1$ , where the function  $g^{(1+B)}(\cdot, \gamma)$  is the  $(1+B)$ -th iterate of  $g(\cdot; \gamma)$ .

Based on Theorem 1, one can efficiently calculate the fixed point  $\tilde{\mathbf{s}}$  given the resource configuration  $(x, y)$  and the average task generating rate  $\bar{\lambda}^{[M]}$ . Accordingly, we often use  $\tilde{\mathbf{s}}(x, y, \bar{\lambda}^{[M]})$  to emphasize the dependency. Recall that this section aims to estimate the time-average workload  $L^{[N]|\delta]}(x, y, \Phi)$  defined in Eq. (4) based on the mean field model. For this goal, we define  $l(x, y, \bar{\lambda}^{[M]})$  based on the fixed point as follows

$$l(x, y, \bar{\lambda}^{[M]}) \triangleq \sum_{i=1}^B \tilde{s}_i(x, y, \bar{\lambda}^{[M]}). \quad (21)$$

As we will show later,  $l(x, y, \bar{\lambda}^{[M]})$  is an accurate estimation of  $L^{[N]|\delta]}(x, y, \Phi)$ . Although neither the fixed point  $\tilde{\mathbf{s}}$  nor  $l(x, y, \lambda)$  has a closed-form expression in general, Corollary 1 presents a closed-form expression of  $l(x, y, \lambda)$  in a special case. We provide the proof in our technical report [44].

*Corollary 1*: Suppose that  $(d, b) = (1, 1)$ , then the fixed point  $\tilde{\mathbf{s}}$  is given by

$$\begin{cases} s_0 = 1, \\ s_i = \rho^i - \frac{\rho^{B+1}}{\sum_{k=0}^B \rho^k} \sum_{k=0}^{i-1} \rho^k, \quad \forall i \in \{1, 2, \dots, B\}, \\ s_{B+1} = 0, \end{cases} \quad (22)$$

where  $\rho \triangleq \bar{\lambda}^{[M]} / (x + y)$ . Moreover, we have

$$l(x, y, \bar{\lambda}^{[M]}) = \frac{1}{1-\rho} + B - \frac{1+B}{1-\rho^{1+B}}. \quad (23)$$

Corollary 1 leads to the following two implications.

- First, the expression in Eq. (23) is the same as the average queue length in an M/M/1/B queuing system with the arrival rate  $\bar{\lambda}^{[M]}$  and the serving rate  $x + y$ .
- Second, JSQ(1) implies that each source will randomly select a passive-mode edge server,<sup>2</sup> and LQF(1)

<sup>2</sup>The case of JSQ(1) could capture the phenomenon where the moving mobile devices (i.e., sources) migrate the computing task to the closest edge servers without inquiring the number of waiting tasks.

implies that the active-mode edge server assists each passive-mode edge server with equal probability. In this case, the  $N$  passive-mode edge servers can be equivalently viewed as  $N$  independent M/M/1/B queuing systems under the migration parameters  $(d, b) = (1, 1)$ .

Therefore, in the case  $(d, b) = (1, 1)$ , the mean field model in Definition 1 provides an perfect approximation or representation for the stochastic system on average. A natural question is how about the cases  $(d, b) \neq (1, 1)$ ? We will answer this question via the following convergence results.

2) *Convergence*: Theorem 2 presents the relationship between the original stochastic system and the deterministic mean field model. Proof is given in our technical report [44].

*Theorem 2: Given the edge server provision  $(\theta, \eta)$  for the  $M$  sources, the limiting system (i.e.,  $M \rightarrow \infty$ ) satisfies*

$$\lim_{\substack{M \rightarrow \infty \\ N=M\theta}} \lim_{\delta \rightarrow \infty} \|L^{[N|\delta]}(x, y, \Phi) - l(x, y, \bar{\lambda}^{[M]})\| = 0, \quad (24)$$

where  $\delta$  is the duration of a single period.

Theorem 2 shows that the mean field model provides an accurate approximation of the time-average workload if the system size  $M$  and the period duration  $\delta$  are large. The intuitions are two-fold:

- The original stochastic system is characterized by a CTMC  $\{S^{[N]}(\tau) : \tau \in [0, \delta]\}$  in each period. As the period duration  $\delta$  increases, the CTMC is approaching to its steady state.
- The drift condition in the mean field model is defined based on the expected drift of the original stochastic system. As  $N$  increases, the CTMC  $\{S^{[N]}(\tau) : \tau \in [0, \delta]\}$  will behave closer to its expectation (i.e., the mean field model) by the Law of large number [30].

So far, we have introduced the connection between the deterministic mean field model and the stochastic edge network. Although the above analysis assumes that the computing intensity follows the exponential distribution, one can obtain similar results under the general distributions with decreasing hazard rate based on *asymptotic independence* or *propagation of chaos* [45]. We refer interested readers to Section 9.1 of [26] for more details.

## V. AN ONLINE MEAN FIELD POLICY

This section proposes an *online mean field aided configuration policy*  $\mathfrak{A}$ , which leverages the mean field model to address the partial feedback issue in Problem 1. To proceed, we first define the *mean field cost* of each period  $t$  as follows

$$C_t(z_t, \bar{\lambda}_t^{[M]}) \triangleq G\left(l(z_t, \bar{\lambda}_t^{[M]})\right) + \xi_t^P \theta x_t^\sigma + \xi_t^A \eta y_t^\sigma, \quad (25)$$

which replaces the time-average workload  $L_t^{[N|\delta]}(z_t, \Phi_t)$  with the deterministic formula  $l(z_t, \bar{\lambda}_t^{[M]})$ . Moreover, the mean field cost in Eq. (25) has the following features.

- Theorem 2 implies that the mean field cost in Eq. (25) is an accurate estimation for the operator's real cost in Eq. (9).
- At the end of period  $t$ , the operator can efficiently compute the mean field cost for any configuration  $z \in \mathcal{Z}$

after observing the average task generating rate  $\bar{\lambda}_t^{[M]}$  and the per-unit expenditure  $(\xi_t^P, \xi_t^A)$ .

The mean field cost exhibits an explicit expression in Eq. (25), however, one can check that it is non-convex in  $z_t$ , and the gradient may not be bounded. Therefore, the gradient-based online convex optimization algorithms (e.g., OGD [40]) cannot preserve the no-regret performance in our problem. Moreover, the learning algorithms (e.g., [41]–[43]) that rely on Lipschitz condition do not work here. To overcome these challenges, we will propose our approach in Section V-A and analyze the performance in Section V-B.

### A. Online Mean Field Aided Policy

1) *Basic Idea*: Our proposed policy  $\mathfrak{A}$  works in two phases: (a) *discretization phase* and (b) *learning phase*. In the discretization phase, we discretize the metric space  $\mathcal{Z}$  into a finite set  $\mathcal{A}$  of resource candidates. In the learning phase, we learn about the optimal resource candidate based on the mean field model, which enables us to obtain the full feedback in the learning phase. Moreover, we will show that the discretization phase only incurs a bounded performance loss. We summarize the proposed policy  $\mathfrak{A}$  in Algorithm 1 and elaborate it in the following.

2) *Discretization Phase*: We discretize the metric space  $\mathcal{Z}$  based on a parameter  $\beta > 0$ , i.e., Line 1 and Line 2 in Algorithm 1. Specifically, for the passive-mode edge servers, we define  $x_{[i]}$  as follows

$$x_{[i]} \triangleq \min(x_H, x_L(1 + \beta)^i), \quad \forall i \in \{1, 2, \dots, A_x\}, \quad (26)$$

where  $A_x \triangleq \lceil \log_{1+\beta} \frac{x_H}{x_L} \rceil$  depends on the parameter  $\beta$  and  $\lceil \cdot \rceil$  is the ceil function. Note that the set  $\{x_{[i]} : \forall 1 \leq i \leq A_x\}$  includes all the powers of  $1 + \beta$  in the range  $[x_L, x_H]$ . Similarly, for the active-mode edge servers, we define  $y_{[j]}$  as follows:

$$y_{[j]} \triangleq \min(y_H, y_L(1 + \beta)^j), \quad \forall j \in \{1, 2, \dots, A_y\}, \quad (27)$$

where  $A_y \triangleq \lceil \log_{1+\beta} \frac{y_H}{y_L} \rceil$ .

Based on the discussions above, the set  $\mathcal{A} \triangleq \{(i, j) : \forall 1 \leq i \leq A_x, 1 \leq j \leq A_y\}$  contains all the resource configuration candidates after discretizing  $\mathcal{Z}$  based on the parameter  $\beta$ . We provide an illustrative example in Fig. 2 where we obtain the red crosses (i.e.,  $\mathcal{A}$ ) from the shaded region (i.e.,  $\mathcal{Z}$ ) with  $A_x = A_y = 5$ . Note that a smaller parameter  $\beta$  leads to a more precise discretization, thus a larger set  $\mathcal{A}$ . In Section V-B, we will show how the parameter  $\beta$  affects the theoretical performance of the proposed policy  $\mathfrak{A}$  in Algorithm 1. Next we introduce the learning phase based on the set  $\mathcal{A}$ .

3) *Learning Phase*: The learning phase of policy  $\mathfrak{A}$  follows the multiplicative weight update (MWU) method [31]. Specifically, we maintain a weight matrix  $w_t = \{w_t(i, j) \in [0, 1] : \forall (i, j) \in \mathcal{A}\}$  in each period  $t$ . As we will see later, the weight  $w_t(i, j)$  in period  $t$  is negatively related to the total mean field cost incurred by the candidate  $(x_{[i]}, y_{[j]})$  before period  $t$ . Hence a larger weight corresponds to a better resource configuration. Moreover, our proposed policy  $\mathfrak{A}$  uses the weight matrix  $w_t$  to calculate the probabilistic selection

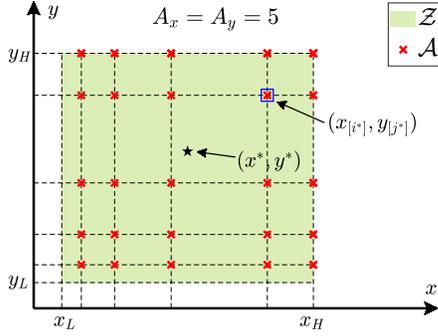


Fig. 2. An illustration for proposed policy  $\mathfrak{Q}$ .

matrix  $\mathbf{p}_t = \{p_t(i, j) : \forall (i, j) \in \mathcal{A}\}$ , and determines the resource configuration probabilistically based on  $\mathbf{p}_t$ . Overall, the resource configuration candidate with a larger weight is selected with a higher probability.

As shown in Algorithm 1, the learning phase includes Lines 3~9. Specifically, policy  $\mathfrak{Q}$  will initialize the weight matrix equally and repeat the following two steps:

- **Line 5~Line 6:** We calculate the probabilistic selection matrix  $\mathbf{p}_t$  based on the weight matrix  $\mathbf{w}_t$  as follows:

$$p_t(i, j) \triangleq \frac{w_t(i, j)}{\sum_{(i', j') \in \mathcal{A}} w_t(i', j')}, \quad \forall (i, j) \in \mathcal{A}. \quad (28)$$

We then determine the resource configuration  $(x_{[i_t]}, y_{[j_t]})$  in period  $t$  by randomly drawing a tuple  $(i_t, j_t) \in \mathcal{A}$  according to the probability distribution  $\mathbf{p}_t$ .

- **Line 7~Line 9:** At the end of period  $t$ , the operator observes the average task generating rate  $\bar{\lambda}_t^{[M]}$ , and calculates the normalized mean field cost  $c_t(i, j)$  based on the mean field model as follows:

$$c_t(i, j) \triangleq \frac{C_t(x_{[i]}, y_{[j]}, \bar{\lambda}_t^{[M]})}{\bar{C}}, \quad \forall (i, j) \in \mathcal{A}. \quad (29)$$

where  $\bar{C} \triangleq G(B) + x_H^{\sigma} c_H^P + y_H^{\sigma} c_H^A$  represents the potential maximal mean field cost. Finally, we update the weight matrix  $\mathbf{w}_{t+1}$  for the next period:

$$w_{t+1}(i, j) = w_t(i, j) \cdot (1 - \epsilon)^{c_t(i, j)}, \quad \forall (i, j) \in \mathcal{A}. \quad (30)$$

where  $\epsilon \in (0, 1)$  is initialized in Line 1.

So far we have introduced the proposed policy  $\mathfrak{Q}$  in Algorithm 1. Next we move on to the performance analysis.

### B. Performance Analysis

Recall that Theorem 2 unveils the convergence relationship between the original stochastic edge network and the deterministic mean field model. Hence our performance analysis in this section will focus on the mean field cost. Specifically, the total mean field cost incurred by our proposed policy  $\mathfrak{Q}$  is

$$C_T^{\mathfrak{Q}} = \sum_{t=1}^T \mathbb{E} \left[ C_t(x_{[i_t]}, y_{[j_t]}, \bar{\lambda}_t^{[M]}) \mid \mathbf{p}_t \right], \quad (31)$$

where the expectation is taken over the randomness in Line 6 of Algorithm 1. We will analyze the performance gap between

### Algorithm 1: Mean Field Aided Policy $\mathfrak{Q}$

**Output:** Decisions  $(x_{[i_t]}, y_{[j_t]})$  for each period  $t$ .

- 1 **Initial**  $\epsilon \in (0, 1)$  and  $\beta > 0$ .
- 2 **Define** the resource candidate  $(x_{[i]}, y_{[j]})$  for any tuple  $(i, j) \in \mathcal{A}$  based on Eq. (26) and Eq. (27).
- 3 **Initial**  $w_1(i, j) = 1$  for any tuple  $(i, j) \in \mathcal{A}$
- 4 **for**  $t = 1$  **to**  $T$  **do**
- 5     **Calculate** the selection probability distribution  $\mathbf{p}_t = \{p_t(i, j) : \forall (i, j) \in \mathcal{A}\}$  according to Eq. (28)
- 6     **Determine**  $(x_{[i_t]}, y_{[j_t]})$  by randomly drawing a tuple  $(i_t, j_t) \in \mathcal{A}$  according to the distribution  $\mathbf{p}_t$
- 7     **Calculate** the mean field cost  $C_t(x_{[i]}, y_{[j]}, \bar{\lambda}_t^{[M]})$  for any tuple  $(i, j) \in \mathcal{A}$ .
- 8     **Calculate** the normalized mean-field cost  $c_t(i, j)$  for any  $(i, j) \in \mathcal{A}$  based on Eq. (29)
- 9     **Update** weight matrix  $\mathbf{w}_{t+1}$  according to Eq. (30)
- 10 **end**

the proposed policy  $\mathfrak{Q}$  and the offline optimal configuration  $(x^*, y^*)$ , defined as

$$(x^*, y^*) \triangleq \arg \min_{(x, y) \in \mathcal{Z}} \sum_{t=1}^T C_t(x, y, \bar{\lambda}_t^{[M]}). \quad (32)$$

Accordingly, we let  $C_T^* \triangleq \sum_{t=1}^T C_t(x^*, y^*, \bar{\lambda}_t^{[M]})$  denote the offline minimal mean field cost.

The performance gap between  $C_T^*$  and  $C_T^{\mathfrak{Q}}$  depends on the two parameters  $\epsilon$  and  $\beta$  in Algorithm 1. Roughly speaking, the performance loss of our proposed policy  $\mathfrak{Q}$  consists of the *discretization loss* and the *learning loss*, which are presented in Lemma 1 and Lemma 2, respectively. We provide the proofs in our technical report [44].

*Lemma 1 (Discretization Loss):* There exists a tuple  $(i^*, j^*) \in \mathcal{A}$  satisfying the following conditions

$$x_{[i^*-1]} \leq x^* \leq x_{[i^*]}, \quad (33a)$$

$$y_{[j^*-1]} \leq y^* \leq y_{[j^*]}. \quad (33b)$$

Moreover, we have

$$\sum_{t=1}^T C_t(x_{[i^*]}, y_{[j^*]}, \bar{\lambda}_t^{[M]}) \leq (1 + \beta)^{\sigma} C_T^*, \quad (34)$$

where  $\sigma \geq 1$  is the exponent of operation expenditure.

Lemma 1 shows that the discretization scheme in policy  $\mathfrak{Q}$  increases at most a constant factor of  $(1 + \beta)^{\sigma}$  compared to the offline minimal cost  $C_T^*$ . The numerical example in Fig. 2 illustrates the relationship in Eq. (33) between  $(i^*, j^*)$  and  $(x^*, y^*)$ . Specifically, suppose that the black star represents  $(x^*, y^*)$ , then the resource candidate labeled by a blue square represents  $(x_{[i^*]}, y_{[j^*]})$ . According to Eq. (34), one should adopt a smaller  $\beta > 0$  to reduce the discretization loss. However, as shown by Lemma 2, a smaller parameter  $\beta$  will lead to a larger learning loss, since a smaller  $\beta$  enlarges the set  $\mathcal{A}$ .

*Lemma 2 (Learning Loss):* For any tuple  $(i, j) \in \mathcal{A}$ , we have

$$C_T^{\mathfrak{A}} \leq \Psi(\epsilon, \beta) + \frac{1}{\epsilon} \ln \left( \frac{1}{1-\epsilon} \right) \sum_{t=1}^T C_t(x_{[i]}, y_{[j]}, \bar{\lambda}_t^{[M]}), \quad (35)$$

where  $\sum_{t=1}^T C_t(x_{[i]}, y_{[j]}, \bar{\lambda}_t^{[M]})$  is the total mean field cost incurred by  $(x_{[i]}, y_{[j]})$ . Moreover, the constant  $\Psi(\epsilon, \beta)$  is

$$\Psi(\epsilon, \beta) \triangleq \frac{\bar{C}}{\epsilon} \ln \left( \left( 1 + \frac{\ln \left( \frac{x_H}{x_L} \right)}{\ln(1+\beta)} \right) \left( 1 + \frac{\ln \left( \frac{y_H}{y_L} \right)}{\ln(1+\beta)} \right) \right). \quad (36)$$

Lemma 2 indicates that the learning loss of policy  $\mathfrak{A}$  jointly depends on the two parameters  $\epsilon$  and  $\beta$ . Note that the constant  $\Psi(\epsilon, \beta)$  decreases in  $\epsilon$ , while the coefficient  $\frac{1}{\epsilon} \ln \left( \frac{1}{1-\epsilon} \right)$  increases in  $\epsilon$ . Hence there is a trade-off in choosing the parameter  $\epsilon$ . Moreover, the constant  $\Psi(\epsilon, \beta)$  decreases in the parameter  $\beta$ , thus a smaller  $\beta$  reduces the discretization loss, but increases the learning loss.

Theorem 3 presents the relationship between  $C_T^{\mathfrak{A}}$  and  $C_T^*$  by combining Lemma 1 and Lemma 2.

*Theorem 3:* For any  $\epsilon \in (0, 1)$  and  $\beta > 0$ , the policy  $\mathfrak{A}$  in Algorithm 1 attains the following performance

$$C_T^{\mathfrak{A}} \leq \Psi(\epsilon, \beta) + \frac{1}{\epsilon} \ln \left( \frac{1}{1-\epsilon} \right) (1+\beta)^\sigma C_T^*. \quad (37)$$

Theorem 3 shows that the expected mean field cost incurred by the policy  $\mathfrak{A}$  is no greater than the sum of a constant multiple of the offline minimal cost  $C_T^*$  and an extra constant  $\Psi(\epsilon, \beta)$ . To have a better understanding on Theorem 3, we will illustrate how to set the two parameters  $\epsilon$  and  $\beta$  from different angles in Corollary 2 and Corollary 3, respectively.

*Corollary 2:* For any  $\alpha \in (0, 2^{\sigma+1}]$ , suppose that the offline minimal mean field cost satisfies  $C_T^* \geq C_{thr}$ , where  $C_{thr}$  is

$$C_{thr} \triangleq \frac{4^{1+\sigma} \bar{C}}{\alpha^2} \left[ 2 \ln \left( \frac{2^{2+\sigma}}{\alpha} \right) + J \right], \quad (38)$$

and the constant  $J \triangleq \ln \left( \ln \frac{2x_H}{x_L} \right) + \ln \left( \ln \frac{2y_H}{y_L} \right) + 2 \ln(2)$ . Let  $\epsilon = \beta = \frac{\alpha}{2^{1+\sigma}}$ , then our proposed policy  $\mathfrak{A}$  in Algorithm 1 attains the following performance

$$C_T^{\mathfrak{A}} \leq (1+\alpha) C_T^*. \quad (39)$$

Corollary 2 shows that policy  $\mathfrak{A}$  is  $(1+\alpha)$ -competitive if the inequality condition  $C_T^* \geq C_{thr}$  holds. Specifically, this condition requires that the offline minimal mean field cost  $C_T^*$  should be no smaller than the threshold  $C_{thr}$ , which does not scales in  $T$ . Recall that  $C_T^*$  corresponds to the offline optimal result across the  $T$  periods, thus it would linearly increase in  $T$ . This means that the condition in Eq. (38) will automatically hold when the number of periods  $T$  is large, thus it is a mild condition.

*Corollary 3:* With the parameters  $\epsilon = \sqrt{\frac{\ln(T)}{T}}$  and  $\beta = \frac{1}{\sqrt{T}}$ , our proposed policy  $\mathfrak{A}$  in Algorithm 1 achieves

$$C_T^{\mathfrak{A}} - C_T^* \leq \bar{C} \sqrt{T \ln(T)} \left[ 2^{1+\sigma} + \frac{J}{\ln(T)} \right], \quad (40)$$

where  $J \triangleq \ln \left( \ln \frac{2x_H}{x_L} \right) + \ln \left( \ln \frac{2y_H}{y_L} \right) + 2 \ln(2)$ .

Corollary 3 presents another way of setting the two parameters  $\epsilon$  and  $\beta$ . We elaborate this corollary from two aspects:

- First, the inequality (40) indicates that our proposed policy  $\mathfrak{A}$  attains a sub-linear regret in the order

$O(\sqrt{T \ln(T)})$ . This means that the resource configuration decisions  $\{(x_{[i]}, y_{[j]}) : \forall t \in \mathcal{T}\}$  generated by our proposed policy  $\mathfrak{A}$  performs asymptotically as well as the offline optimal configuration  $(x^*, y^*)$ .

- Second, the regret order  $O(\sqrt{T \ln(T)})$  is slightly greater than the well-known lower bound  $O(\sqrt{T})$  for the regret in online convex optimization (e.g., [40]).<sup>3</sup> As far as we know, the learning algorithm proposed by Yang *et al.* in [43] matches such a lower bound in the general Lipschitz expert problem [42]. However, it relies on the Lipschitz condition, which is not available in our problem. To sum up, Corollary 3 indicates that our proposed policy  $\mathfrak{A}$  requires less idealistic condition (i.e., Lipschitz condition) at the price of slightly increasing the regret compared to the lower bound by  $O(\sqrt{\ln(T)})$ .

## VI. NUMERICAL RESULTS

We carry out extensive evaluation on the proposed policy  $\mathfrak{A}$  based on some empirical data. We will consider a hybrid edge server provision mode  $(\theta, \eta) = (0.5, 0.1)$  and investigate the impact of network scale (i.e., the number of sources  $M$ ).<sup>4</sup> Specifically, we will start with the single-period demonstration in Section VI-A, and then carry out the multi-period evaluation in Section VI-B.

### A. Single-Period Demonstration

We consider a single period and compare the mean field model to the stochastic edge network. Specifically, we will fix the resource configuration  $(x, y) = (1, 5)$  and the Poisson task generating rate  $\lambda^m = 0.9$  for each source  $m \in \mathcal{M}$ .

Fig. 3(a) shows how the mean field model  $s(\tau)$  evolves from the initial state  $s^0 = \mathbf{0}$ . The vertical axis represents the average workload  $\sum_{i=1}^B s_i(\tau)$  and the three curves correspond to different migration parameters  $(d, b)$ , respectively. As time  $\tau$  increases, the mean field model converges to the fixed point  $\tilde{s}$  defined in Theorem 1.

Fig. 3(b)-3(d) plot the average workload  $L^{[N]}(\tau, x, y, \Phi)$  of the stochastic edge network, and the three sub-figures correspond to different migration parameters, respectively. In each sub-figure, the three solid curves represent different numbers of sources, i.e.,  $M \in \{20, 100, 1000\}$ . The black dash curve is the same as that in Fig. 3(a). Note that the three solid curves are all centered on the black dash curve with some fluctuation. Moreover, a larger system size corresponds to a smaller fluctuation. These observations are consistent with the convergence results in Theorem 2.

Comparing the three sub-figures in Fig. 3, we find it significantly reduces the average workload by changing the parameters  $(d, b)$  from  $(1, 1)$  to  $(2, 2)$ . However, it merely leads to a tiny reduction by further increasing to  $(3, 3)$ . This means that the migration schemes JSQ(2) and LQF(2) slightly

<sup>3</sup>Note that our problem with the mean field feedback corresponds to the online non-convex learning, which is more general than the online convex optimization (OCO). Hence the regret lower bound in OCO still holds in our problem.

<sup>4</sup>Given the hybrid edge server provision  $(\theta, \eta) = (0.5, 0.1)$ , every  $M = 10$  sources correspond to  $N = 5$  passive-mode edge servers and  $K = 1$  active-mode edge server.

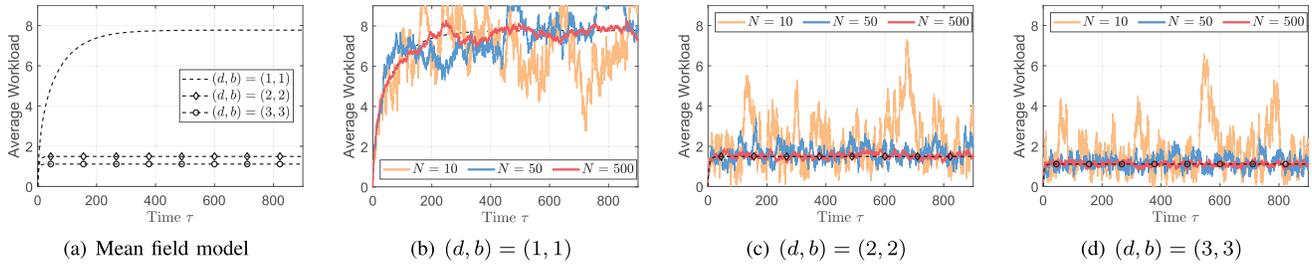


Fig. 3. Average workload in the stochastic edge network.

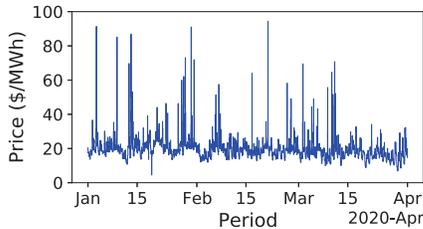


Fig. 4. Electricity prices.

increase the communication overhead, but can reduce the average workload considerably. We will verify this claim under the dynamic setting with multiple periods in Section VI-B.

### B. Multi-Period Evaluation

We evaluate the proposed policy  $\mathfrak{A}$  based on the real world electricity market prices in US [46]. Fig. 4 plots the hourly prices of the first three months in 2020. We consider a total of  $T = 2208$  periods (i.e., hours), and let  $p_t$  denote the electricity price in period  $t$ . As for the operation expenditure, we consider  $\sigma = 1$ , and quantify the per-unit operation expenditure according to  $\xi_t^P = 0.1p_t$  and  $\xi_t^A = 0.08p_t$ . Furthermore, we first generate the sources' task generation rate in different periods according to a uniform distribution on the support  $[0, 1]$ . We then randomly generate the task profile  $\Phi_t$  accordingly. We consider the feasible resource range  $[0.2, 2]$  for each edge server, and use  $G(l) = 30l$  to measure the performance degradation. We evaluate the proposed policy  $\mathfrak{A}$  with the parameters  $(\epsilon, \beta) = (0.2, 0.5)$  and compare the following three cases:

- Case Mfm corresponds to the mean field model and measures the mean field cost incurred by Algorithm 1.
- Case Alg( $N$ ) corresponds to the original stochastic edge computing network with  $N = \theta M$  passive-mode edge servers, and measures the real operating cost incurred by Algorithm 1.
- Case Bch( $N$ ) is the benchmark scheme of case Alg( $N$ ) and does not rely on the mean field model. That is, Bch( $N$ ) adopts the same discretization phase as in Algorithm 1, but only relies on the observed partial feedback in the learning phase.

We run the evaluation for one hundred times and visualize the results in Fig. 5. The three sub-figures correspond to different migration parameters, i.e.,  $(d, b) = (1, 1)$ ,  $(d, b) = (2, 2)$ , and  $(d, b) = (3, 3)$ . In each sub-figure, the black

dash line represents the time-average offline minimal mean field cost, i.e.,  $C_T^*/T$ . The black circle curve corresponds to the case Mfm. The blue and red curves with markers represent case Alg(5) and case Alg(10), respectively. The blue and red curves without marker represent case Bch(5) and case Bch(10), respectively. The shaded region represents the three-sigma regime over the one hundred simulation runs. We have the following observations from each sub-figure in Fig. 5:

- **Performance of Proposed Policy:** The black circle (i.e., Mfm) curve gradually converges to the black dash line, which verifies the asymptotic optimality of our proposed policy  $\mathfrak{A}$  in Algorithm 1.
- **Accuracy of Mean Field Model:** Both the blue triangle curve (i.e., case Alg(5)) and the red square curve (i.e., case Alg(10)) have a slight difference compared to black circle curve (i.e., case Mfm). This is due to the mean field approximation gap. Note that the red square curve (i.e., case Alg(10)) is even closer to the black curve than the blue triangle curve (i.e., Alg(5)). This observation shows that the approximation gap decreases in the system size, which is consistent with the convergence result in Theorem 2.
- **Advantage of Mean Field Model:** Comparing the two blue curves shows that case Bch(5) would take much longer time to converge to the offline optimal outcome  $C_T^*/T$  than case Alg(5). The two red curves also lead to a similar observation. These observations indicate that the mean field model can significantly speed up the convergence to the offline optimal outcome in the online resource configuration problem.

By comparing the three sub-figures in Fig. 5, we note that the larger migration parameters  $(d, b)$  lead to a smaller operating cost. This motivates us to investigate the impact of  $(d, b)$ . Fig. 6 investigates the influence of the migration parameters focusing on case Mfm and case Alg(10). Specifically, Fig. 6 plots the time-average performance degradation (PeDe) and the operation expenditure (OpEx) over the  $T$  periods. At the top of each bar, we label the percentage of the cost reduction compared to  $(d, b) = (1, 1)$ . Fig. 6 leads to the following two-fold implications:

- First, increasing the migration parameters  $(d, b)$  can reduce the operator's total operating cost up to 45%. In this progress, however, the communication overhead also increases linearly in the two parameters  $d$  and  $b$ .

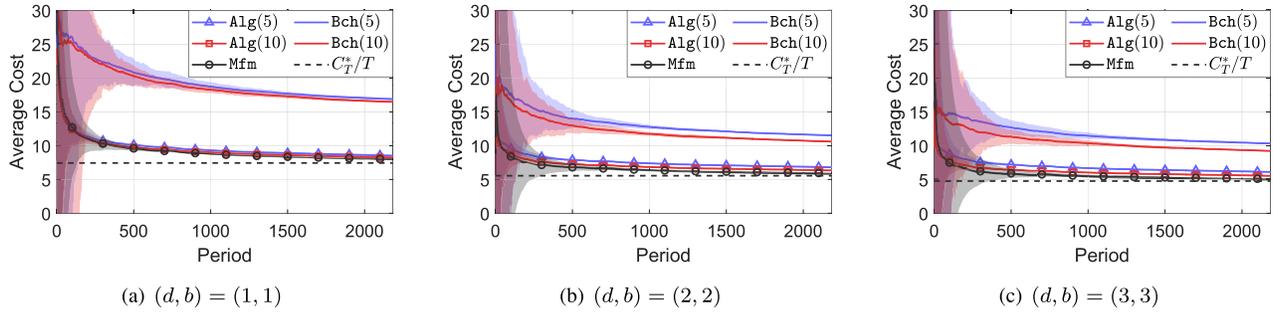


Fig. 5. Time-average operating cost in cases Mfm, Alg( $N$ ), and Bch( $N$ ).

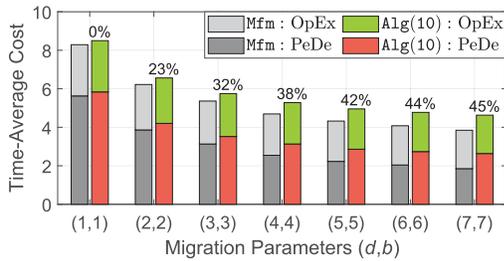


Fig. 6. Impact of parameters in JSQ( $d$ ) and LQF( $b$ ).

- Second, the cost reduction when  $(d, b) = (2, 2)$  is already greater than half of that when  $(d, b) = (7, 7)$ .

The above observations show that the migration schemes JSQ(2) and LQF(2) under our proposed policy  $\mathfrak{A}$  can considerably reduce the operating cost with a small increase in the communication overhead.

## VII. CONCLUSION

In this paper, we focus on the large-scale hybrid edge server provision under two decentralized task migration schemes, and study how to configure the computing resource in an online dynamic fashion. Specifically, the dynamic resource configuration of a large-scale stochastic edge network corresponds to an online cost minimization problem with the partial feedback issue. To address the partial feedback, we derive a deterministic mean field model, which enables us to estimate the performance of the stochastic system. Based on the mean field model, we propose an online mean field aided configuration policy. Our analysis shows that the proposed policy can attain a constant competitive ratio or a sub-linear regret under different policy parameters. That is, our proposed policy asymptotically performs as well as the offline optimal configuration. To our knowledge, we are the first to integrate mean field theory and online learning with partial feedback. We believe that our results in this paper can improve the efficiency of edge server provision and facilitate the large-scale implementation.

## REFERENCES

- [1] Z. Wang, J. Ye, and J. C. S. Lui, "An online mean field approach for hybrid edge server provision," in *Proc. 22nd Int. Symp. Theory, Algorithmic Found., Protocol Design Mobile Netw. Mobile Comput.*, Jul. 2021, pp. 131–140.
- [2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
- [3] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
- [4] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-intelligence-enabled intelligent 6G networks," *IEEE Netw.*, vol. 34, no. 6, pp. 272–280, Nov./Dec. 2020.
- [5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.
- [6] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
- [7] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Sep. 2016.
- [8] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 207–215.
- [9] I. Hou, T. Zhao, S. Wang, and K. Chan, "Asymptotically optimal algorithm for online reconfiguration of edge-clouds," in *Proc. ACM MobiHoc*, 2016, pp. 291–300.
- [10] W. C. Ng *et al.*, "Stochastic coded offloading scheme for unmanned aerial vehicle-assisted edge computing," *IEEE Internet Things J.*, early access, Feb. 10, 2022, doi: [10.1109/JIOT.2022.3150472](https://doi.org/10.1109/JIOT.2022.3150472).
- [11] A. Asheralieva, D. Niyato, and Z. Xiong, "Auction-and-Learning based Lagrange coded computing model for privacy-preserving, secure, and resilient mobile edge computing," *IEEE Trans. Mobile Comput.*, early access, Jul. 16, 2021, doi: [10.1109/TMC.2021.3097380](https://doi.org/10.1109/TMC.2021.3097380).
- [12] Y. Sahni, J. Cao, and L. Yang, "Data-aware task allocation for achieving low latency in collaborative edge computing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3512–3524, Apr. 2018.
- [13] A. Galanopoulos, T. Salonidis, and G. Iosifidis, "Cooperative edge computing of data analytics for the Internet of Things," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 4, pp. 1166–1179, Dec. 2020.
- [14] M. Tang, L. Gao, and J. Huang, "Enabling edge cooperation in tactile internet via 3C resource sharing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2444–2454, Nov. 2018.
- [15] A. Ndikumana *et al.*, "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1359–1374, Jun. 2020.
- [16] K. Peng *et al.*, "Joint optimization of service chain caching and task offloading in mobile edge computing," *Appl. Soft Comput.*, vol. 103, May 2021, Art. no. 107142. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S156849462100065X>
- [17] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 10–18.
- [18] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1094–1104, Oct. 2001.
- [19] L. Ying, R. Srikant, and X. Kang, "The power of slightly more than one sample in randomized load balancing," *Math. Oper. Res.*, vol. 42, no. 3, pp. 692–722, Aug. 2017.

- [20] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.
- [21] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2082–2091, Dec. 2017.
- [22] J. Shyuan, W. Lim, Z. Xiong, T. D. Niyato, C. Leung, and C. Miao, "A double auction mechanism for resource allocation in coded vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1832–1845, Feb. 2022.
- [23] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.
- [24] E. Meskar and B. Liang, "Fair multi-resource allocation in mobile edge computing with multiple access points," in *Proc. ACM MobiHoc*, 2020, pp. 11–20.
- [25] Z. Zhou, X. Chen, W. Wu, D. Wu, and J. Zhang, "Predictive online server provisioning for cost-efficient iot data streaming across collaborative edges," in *Proc. ACM MobiHoc*, 2019, pp. 321–330.
- [26] M. van der Boor, S. C. Borst, J. S. H. van Leeuwen, and D. Mukherjee, "Scalable load balancing in networked systems: A survey of recent advances," 2018, *arXiv:1806.05444*.
- [27] M. Bramson, Y. Lu, and B. Prabhakar, "Randomized load balancing with general service time distributions," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 275–286, Jun. 2010.
- [28] Q. Xie, X. Dong, Y. Lu, and R. Srikant, "Power of d choices for large-scale bin packing: A loss model," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 43, no. 1, pp. 321–334, 2015.
- [29] J. N. Tsitsiklis and K. Xu, "On the power of (Even a Little) resource pooling," *Stochastic Syst.*, vol. 2, no. 1, pp. 1–66, Jun. 2012.
- [30] L. Ying, "On the approximation error of mean-field models," *Stochastic Syst.*, vol. 8, no. 2, pp. 126–142, Jun. 2018.
- [31] S. Arora and S. Kale, "The multiplicative weights update method: A meta-algorithm and applications," *Theory Comput.*, vol. 8, no. 1, pp. 121–164, May 2012.
- [32] V. Vovk, "A game of prediction with expert advice," *J. Comput. Syst. Sci.*, vol. 56, no. 2, pp. 153–173, Apr. 1998.
- [33] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Mar. 2018.
- [34] A. Reiszadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, "Coded computation over heterogeneous clusters," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4227–4242, Jul. 2019.
- [35] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. Cambridge, MA, USA: MIT Press, 1965.
- [36] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.
- [37] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
- [38] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Aug. 2016.
- [39] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1431–1439.
- [40] E. Hazan, "Introduction to online convex optimization," *Found. Trends Optim.*, vol. 2, nos. 3–4, pp. 157–325, 2016.
- [41] R. Kleinberg and A. Slivkins, "Sharp dichotomies for regret minimization in metric spaces," in *Proc. SODA*. Philadelphia, PA, USA: SIAM, 2010, pp. 827–846.
- [42] R. Kleinberg, A. Slivkins, and E. Upfal, "Bandits and experts in metric spaces," *J. ACM*, vol. 66, no. 4, pp. 1–77, Aug. 2019, doi: 10.1145/3299873.
- [43] L. Yang, L. Deng, M. H. Hajiesmaili, C. Tan, and W. S. Wong, "An optimal algorithm for online non-convex learning," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 2, no. 2, pp. 1–25, Jun. 2018, doi: 10.1145/3224420.
- [44] *Technical Report*. Accessed: Mar. 10, 2022. [Online]. Available: <https://wang-zhi-yuan.github.io/file/JSAC22.pdf>
- [45] M. Bramson, Y. Lu, and B. Prabhakar, "Asymptotic independence of queues under randomized load balancing," *Queueing Syst.*, vol. 71, no. 3, pp. 247–292, Jul. 2012.
- [46] *PJM*. Accessed: Jul. 31, 2020. [Online]. Available: [https://dataminer2.pjm.com/feed/rt\\_hrl\\_lmps/definition](https://dataminer2.pjm.com/feed/rt_hrl_lmps/definition)



**Zhiyuan Wang** received the B.Eng. degree in information engineering from Southeast University, Nanjing, in 2016, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong in 2019. He was a Post-Doctoral Fellow with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, from 2019 to 2021. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University. His research interests include game theory, mean field theory, and online learning, with current emphasis on edge computing and age of information.



**Jiancheng Ye** (Member, IEEE) received the B.E. degree in network engineering from Sun Yat-sen University, China, in 2008, the M.Phil. degree in computer science and engineering from The Hong Kong University of Science and Technology in 2011, and the Ph.D. degree in computer networking from The University of Hong Kong in 2018. From 2011 to 2014, he was a Software Engineer with Harmonic Inc. He was a Post-Doctoral Fellow with The University of Hong Kong from 2018 to 2019. He is currently a Researcher with the Network Technology Laboratory, Huawei, Hong Kong. His research interests include congestion control, queue management, optimization of computer networks, edge computing, and online learning. He is a member of ACM.



**John C. S. Lui** (Fellow, IEEE) received the Ph.D. degree in computer science from the University of California at Los Angeles. He is currently the Choh-Ming Li Chair Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His current research interests include machine learning, online learning (e.g., multi-armed bandit and reinforcement learning), network science, future internet architectures and protocols, network economics, network/system security, and large scale storage systems. He is an Elected Member of the IFIP WG 7.3, a fellow of ACM, and a Senior Research Fellow of the Croucher Foundation. He has received various departmental teaching awards and the CUHK Vice Chancellor's Exemplary Teaching Award. He was a co-recipient of the Best Paper Award in the IFIP WG 7.3 Performance 2005, IEEE/IFIP NOMS 2006, SIMPLEX 2013, and ACM RecSys 2017. He was the Past Chair of the ACM SIGMETRICS (2011–2015).