Online Orchestration of Cross-Edge Service Function Chaining for Cost-Efficient Edge Computing

Zhi Zhou, *Member, IEEE*, Qiong Wu, *Student Member, IEEE*, and Xu Chen, *Member, IEEE*

***Abstract*— Edge computing (EC) has quickly ascended to be the *de-facto* standard for hosting emerging low-latency applications, as exemplified by intelligent video surveillance, Internet of Vehicles, and augmented reality. For EC, service function chaining is envisioned as a promising approach to configure various services in an agile, flexible, and cost-efficient manner. When running on top of geographically dispersed edge clouds, fully unleashing the benefits of service function chaining is, however, by no means trivial. In this paper, we propose an online orchestration framework for cross-edge service function chaining, which aims to maximize the holistic cost efficiency, via jointly optimizing the resource provisioning and traffic routing on-the-fly. This long-term cost minimization problem is difficult since it is NP-hard and involves future uncertain information. To simultaneously address these dual challenges, we carefully combine an online optimization technique with an approximate optimization method in a joint optimization framework, through: 1) decomposing the long-term problem into a series of one-shot fractional problem with a regularization technique and 2) rounding the fractional solution to a near-optimal integral solution with a randomized dependent scheme that preserves the solution feasibility. The resulting online algorithm achieves an outstanding performance guarantee, as verified by both rigorous theoretical analysis and extensive trace-driven simulations.**

***Index Terms*— Edge Computing, Service Function Chaining, Online Optimization.**

I. Introduction

WITH the advancements in 5G communications and Internet-of-Things (IoT), billions of devices (e.g., mobile devices, wearable devices and sensors) are expected to be connected to the Internet, indispensable for a wide variety of IoT applications, ranging from intelligent video surveillance and Internet of Vehicles (IoV) to augmented reality [1]–[5]. As a result of the proliferation of these diverse applications, large volumes of multi-modal data (e.g., audio and video) of physical surroundings are continuously sensed at the device side. To process such a tremendous amount of data streams which typically requires vast computing resources and very low latency, the concept of edge computing (EC) [1] has been recently proposed. As an extension of cloud computing, EC pushes cloud resources and services from the network core to the network edges that are in closer proximity to IoT devices and data sources, leading to significant reduction of end-to-end latency.

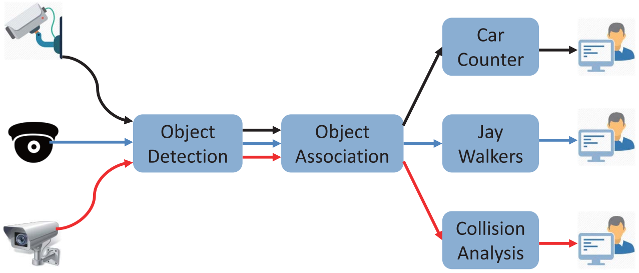

Fig. 1. An example of 3 SFCs for live video analytics, the black, blue and red arrowlines represent the video streams that require the SFC for car counter, jay walkers and collision analysis, respectively.

While recognizing the superiority of edge computing in reducing the user-perceived latency of IoT applications, it is important to note that the low cost-efficiency may become the bottleneck towards sustainable EC ecosystems [6]. Specifically, it is widely acknowledged that the power consumption typically dominates the operational cost of a datacenter, while the power efficiency of an edge node can be hundreds of times less efficient than that of a cloud server [7], as reported by Microsoft. Therefore, efficient cost management is of strategic importance for provisioning sustainable EC service.. To this end, an emerging technology called service function chain (SFC) which provisions services in an agile, flexible and cost-efficient manner has been advocated [8]. With SFC, a EC service is decomposed into a chain of service functions (SFs) or microservices [3] with precedence order, each SF executes a certain function and all SFs are executed by following the specific precedence order. As an illustrative example, Fig. 1 depicts 3 different SFCs for live video analytics which is envisioned as the killer EC application by Microsoft [2]. In this example, since the two SFs object detection and association are the common SFs of the three SFCs, they can be merged by running single instance of the common SFs, allowing for

Manuscript received December 15, 2018; revised June 26, 2019; accepted June 29, 2019. Date of publication July 5, 2019; date of current version August 6, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1001703, in part by the National Science Foundation of China under Grant U1711265 and Grant 61802449, in part by the Program for Guangdong Introducing Innovative and Enterpreneurial Teams under Grant 2017ZT07X355, in part by Guangdong Natural Science Funds under Grant 2018A030313032, and in part by the Fundamental Research Funds for the Central Universities under Grant 17lgjc40. *(Corresponding author: Xu Chen.)*

The authors are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: zhouzhi9@mail.sysu.edu.cn; wuqiong23@mail2.sysu.edu.cn; chenxu35@mail.sysu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2019.2927070

resource sharing among SFCs and leading to significant cost saving.

However, when deploying various SFCs across multiple geographically dispersed edge clouds which have recently gathered great attention [6], [9], fully materializing the benefits of service function chaining is far from trivial, due to the following reasons. First, to fully unleash the potential of resource sharing among multiple SFCs, the traffic may need to traverse various edge clouds that host the corresponding SFs shared by those SFCs. Clearly, due to the geographical distance, cross-edge traffic routing increases the end-to-end latency, and thus *incurs a cost-performance tradeoff that should be judiciously navigated*. Second, as a result of the geo-distribution and time-varying electricity price, the cost of provisioning a SF instance exhibits strong *spatial and temporal variabilities*. Intuitively, such diversities ought to be fully exploited to dynamically provisioning the SF instances, if we want to minimize the cost of running SF instances. Unfortunately, aggressively re-provisioning instances would greatly increase the switching cost caused by launching new SF instances, meaning that we should *strike a nice balance between the instance running and switching costs* when conducting dynamical instance provisioning. Finally, unlike traditional mega-scale cloud datacenters which have abundant resources, the resource volume of an edge cloud is highly limited. As a result, the provisioned instances for each SF can be dispersed at multiple edge clouds, requiring us to carefully *split and route the traffic to those edge clouds, with an awareness to the spatial diversities on both resource cost and performance (in terms of network latency)*.

Keeping the above factors in mind, in this paper, we advocate an online framework to orchestrate service function chaining across multiple edge clouds. Towards the goal of optimizing the holistic cost-efficiency of the cross-edge SFC system, the proposed framework unifies (1) the instances running cost incurred by the energy consumption and the amortized capital expenditure, (2) the instances switching cost due to launching new SF instances, (3) the cloud outsourcing cost due to the usage of the central cloud resources, in case of traffic flash crowd that overweighs the capacity of the edge clouds, and (4) the traffic routing cost that consists of wide-area network (WAN) bandwidth cost and penalty for cross-edge network latency. To minimize the unified cost, the proposed orchestration framework jointly applies two control knobs: (1) dynamical instance provisioning that periodically adapts the number of running SF instances at each edge cloud; and (2) cross-edge traffic routing that tunes the amount of traffic traversed among edge clouds. With the above setup, the cost minimization for cross-edge SFC deployment over the long-term is formulated as a mixed integer linear programming (MILP).

However, for a practical cross-edge system, solving the above MILP is rather difficult, due to the following dual challenges. First, for the instance switching cost incurred every time when launching new SF instances, it couples the instance provisioning decisions over consecutive time periods. As a result, the long-term cost minimization problem is a time-coupling problem that involves future system information. However, in practice, parameters such as traffic arrivals typically fluctuate over time and thus cannot be readily predicted. Then, it is highly desirable to minimize the long-term cost in an online manner, without utilizing the future information as a priori knowledge. Second, even with an offline setting where all the future information is given as a priori knowledge, the corresponding cost minimization problem is proven to be NP-hard. This further complicates the design of an online algorithm that dynamically optimizes the long-term cost based on the historical and current system information.

Fortunately, by looking deep into the structure of the problem, we can address the above dual challenges, by blending the advantages of a regularization method for online algorithm design and a dependent rounding technique for approximation algorithm design. In particular, by applying the regularization technique from the online learning literature, we temporally decompose the relaxed time-coupling problem into a series of one-shot fractional subproblems, which can be exactly solved without requiring any future information. To round the regularized and fractional solution to feasible integer solutions of the original problem, a randomized dependent rounded scheme is carefully designed. The key idea of the rounding scheme is to compensate each rounded-down instance by another rounded-up instance [10]. This ensures that the solution feasibility is maintained without provisioning excessive SF instances, allowing significant improvement on the cost-efficiency. The proposed approximated online algorithm achieves a good performance guarantee, as verified by both rigorous theoretical analysis and extensive simulations based on realistic electricity prices and workload traces.

II. Related Work

As a promising approach to provision computing and network services, service function chaining has gathered great attention from both industrial and academia. A large body of recent research was devoted to exploiting the opportunities or addressing the challenges in deploying service function chains across geo-distributed infrastructures [11].

A majority of the existing literature focused on deploying service function chains across geo-distributed mega-scale datacenters. For example, Fei *et al.* [12] explored how to achieve the goal of load balancing by dynamically splitting workload and assigning service functions to geo-distributed datacenters. Zhou [13] proposed an online algorithm to place service function chains which arrive in an online manner, aiming at maximizing their total value. Abu-Lebdeh *et al.* [14] investigated the problem of minimizing the operational cost without violating the performance requirements, and proposed a tabu search based heuristic to solve the problem. Toward a fair tradeoff between the cost-efficiency and quality of experience (QoE) of the SFC over multi-clouds, Benkacem *et al.* [15] applied the bargaining game theory to jointly optimize the cost and QoE in a balanced manner. A more closely related work is [16], in which Jia *et al.* studied how to jointly optimize instance provisioning and traffic routing to optimize the long-term system wide cost, and proposed an online algorithm with provable performance guarantee. However, our work is different from and complementary to [16]. Problem-wise, we extend

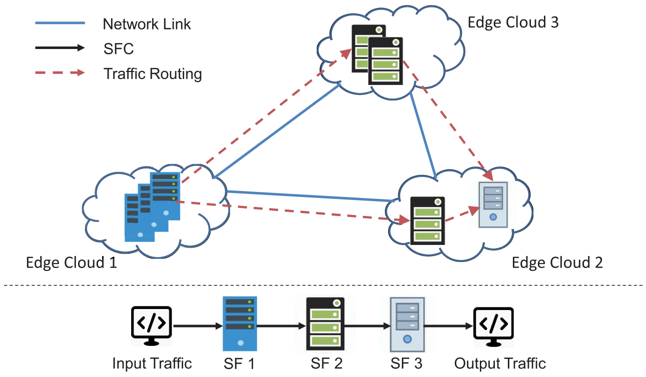

Fig. 2. An illustration of cross-edge service function chain deployment.

the model in [16] by considering: (1) the resource capacity constraints of edge clouds, and (2) a hybrid environment with both resource-limited edge clouds and resource-rich central cloud. Such extensions make the algorithm in [16] not directly applicable to our problem. Algorithm-wise, we incorporate the technique of knapsack cover (KC) constraint to design a more intuitive and simplified online algorithm, yet still preserves the provable performance guarantee.

While significant progress has been made on deploying service function chains across geo-distributed datacenters, the problem in edge computing scenario is only beginning to receive attention. Dinh-Xuan *et al.* [17] proposed heuristic-based placement algorithms that aim to efficiently place the SFC in servers with regard to optimizing service response time and resource utilization. Gouareb *et al.* [18] studied the problem of service function placement and routing across the edge clouds to minimize overall latency, defined as the queuing delay within the edge clouds and in network links. Laghrissi *et al.* [19] developed a spatial-temporal model for service function placement across edge clouds, and compared the performance of several placement strategies in terms of delay and cost. For the emerging paradigm of 5G service-customized network slices, an adaptive interference-aware service function chaining framework [20] is proposed to improve the throughput against performance interference among colocated SFs. Besides, the recent efforts [21]–[24] all adopted an integer linear programming (ILP) formulation to derive computational-efficient heuristics for service function chaining. However, all these works assume a static environment, rather than the stochastic setup considered in this work. Furthermore, all those heuristics except [20] do not provide any performance guarantee.

III. System Model for Cross-Edge SFC Deployment

In this section, we present the system model for cross-edge SFC deployment, and the problem formulation for the long-term cost minimization.

A. Overview of the Cross-Edge System

As illustrated in Fig. 2, we consider an edge service provider running edge computing service on a set of I geographically dispersed edge clouds in proximity to the users (e.g., IoT devices), denoted as $\mathcal{I} = \{1, 2, \ldots, I\}$. Following the recent proposal of architecting edge computing with service function chaining, we assume that each edge computing service performs a series of consecutive service functions (a.k.a, microservice [3]) on the input traffic. For example, for the live video analytics [2] which is envisioned as the killer application of edge computing, an object recognition invokes 3 core vision primitives (i.e., object detection → object association → object classification) to be executed in sequence with the raw input image streams. To characterize such inherent sequential order of service functions, we adopt a service function chain (SFC) with precedence to model the consecutive service functions for each edge computing service [25]. Specifically, we use $\mathcal{M} = \{1, 2, \ldots, M\}$ to denote the set of different SFs that can be selected to form diverse SFCs.

Each SF $m \in \mathcal{M}$ is instantiated in a virtual machine (VM) or container in the edge cloud, the VM or container instances running different SFs are referred to as SF instances. Determined by the resource capacities of the underlying physical servers (e.g., CPU, GPU, memory and network I/O), the service capacity of each instance of SF m in edge cloud i is denoted as b_i^m, meaning the maximal data rate can be supported by an instance of SF m in edge cloud i. Considering the limited amount of resources at each cloud, we further use C_i^m to denote the resource capacity, i.e., maximal number of available instances of SF m in edge cloud i.

The input traffic arrival at each source edge cloud specifies a SFC, and traverses the corresponding SFs that may be deployed across various edge clouds to generate the output result, which is finally returned to the source edge cloud. Specifically, for the input traffic arrival at each source edge cloud $s \in \mathcal{I}$, we denote the requested SFC as SFC s. We further let $h_{mn}^s = 1$ if $m \to n$ is a direct hop of the SFC s (i.e., SF m is the predecessor of SF n in SFC s); and $h_{mn}^s = 0$ otherwise. Note that after the processing of each SF, the traffic rate of each SFC may change at different hops since the SF may increase or decrease the traffic amount. We use α_m^s to denote the change ratio of traffic rate of SFC s on SF m, meaning that the outgoing traffic rate of SFC s after passing an instance of SF m is on average α_m^s times the incoming traffic rate. For ease of presentation, we further use β_m^s to denote the cumulative traffic rate change ratio of SFC s before it goes through SF m, which is the ratio of the overall incoming traffic rate of SF m to the initial total input traffic rate. If we use m_F^s to denote the first SF of SFC s, then we have $\beta_m^s = 1$ if $m = m_F^s$, and $\beta_m^s = \sum_{n \in \mathcal{M}} h_{nm}^s \beta_n^s \alpha_m^s$ if $m \neq m_F^s$.

Without loss of generality, the system works in a time slotted fashion within a large time span of $\mathcal{T} = \{1, 2, \ldots, T\}$. Each time slot $t \in \mathcal{T}$ represents a decision interval, which is much longer than a typical end-to-end delay for the input traffic. At each time slot t, the input traffic rate (in terms of number of data packets per time slot) of each SFC s is denoted as $A_s(t)$. Note that $A_s(t)$ typically fluctuates over time, in the presence of traffic flash crowd at service peak times, the resource required by the input traffic may exceed the overall resource capacity of the cross-edge clouds. To cope with this issue, we assume that a central cloud datacenter with sufficient resource capacity can be leveraged to absorb the extra input traffic.

B. Optimization Space

When running on top of geographically distributed edge clouds, the cross-edge system exhibits strong spatial and temporal variabilities on performance (in terms of the end-to-end latency of the input traffic) and cost. Specifically, at the spatial dimension, the operational cost of running a SF instance fluctuates over time, due to the time-varying nature of the electricity price. While at the temporal dimension, both the operational cost and cross-edge network latency show geographical diversities. Clearly, such diversities ought to be fully exploited if we want to jointly optimize the cost-efficiency of the cross-edge system. Towards this goal, an effective approach is joint dynamical instance provisioning (i.e., dynamically adapting the number of running SF instances at each edge cloud) and cross-edge traffic routing (i.e., dynamically adapting the amount of traffic routed among edge clouds).

We now elaborate the control decisions that we tune to optimize the cost-efficiency of the cross-edge system. First, dynamical instance provisioning, we use $x_i^m(t)$ to denote the number of running instances of SF m provisioned in edge cloud i at time slot t. Since the resource capacity in an edge cloud is highly limited, it is impractical to relax the non-negative integer $x_i^m(t)$ into a real number. Instead, we enforce that $x_i^m(t) \in \{0, 1, 2, \ldots, C_i^m\}$, where C_i^m is the aforementioned maximal number of available instances of SF m in edge cloud i. To efficiently utilize the limited edge cloud resources, an instance of a SF is shared by multiple SFCs whose service chain includes this SF. Second, cross-edge traffic routing, here we use $y_{sij}^{mn}(t)$ to denote the amount of traffic of SFC s, routed from SF m in edge cloud i to SF n in edge cloud j. To denote the routing decisions of the input (output) traffic arrived at (back to) each edge cloud, we further introduce $z_{si}^m(t)$, which represents the total incoming traffic of SFC s to the instances of SF m in edge cloud i. Finally, for ease of problem formulation, we also use $a_s(t)$ and $u_s(t)$ to denote the amount of traffic arrived at each source edge cloud s routed to the edge clouds and the central cloud, respectively.

C. Cost Structure

Given the above control decisions, we are now ready to formulate the overall cost incurred by the cross-edge deployment of the SFCs, which include the instance operation cost, instance switching cost, traffic routing cost and the cloud outsourcing cost.

1) Instance Running Cost: Let $p_i^m(t)$ to denote the cost of running an instance of SF m in edge cloud i and at time slot t, mainly attributed to the power consumption[1] as well as the amortized capital expenditure of the hosting edge server. Then the total instance running cost at time slot t is given by:

$$C_R(t) = \sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} p_i^m(t) x_i^m(t).$$

2) Instance Switching Cost: Launching a new instance of SF m requires transferring a VM image containing the service function to the hosting server, booting it and attaching it to devices on the host server. We use q_i^m to denote the cost of deploying a newly added instance of SF m at edge cloud i. Then the total switching cost at time slot t is given by:

$$C_S(t) = \sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} q_i^m [x_i^m(t) - x_i^m(t-1)]^+,$$

where $[x_i^m(t) - x_i^m(t-1)]^+ = \max\{x_i^m(t) - x_i^m(t-1), 0\}$, denoting the number of newly launched instances of SF m at edge cloud i in time slot t. Note that here we assume that the cost of SF destruction is 0. If this is not the case, we can simply fold the corresponding cost into q_j^m incurred in the next SF launching operation. Without loss of generality, we let $x_i^m(0) = 0$.

3) Cloud Outsourcing Cost: In case of traffic flash crowd whose resource requirement exceeds the capacity of the cross-edge clouds, the extra demand would be outsourced to a remote central cloud with sufficient resource for processing. Here we use r_s to denote the aggregated cost of outsourcing one unit input traffic from the source edge cloud $s \in \mathcal{M}$ to the remote central cloud. The cost r_s includes the resource usage cost of the remote cloud, the bandwidth usage cost and the performance cost incurred by the delay of the WAN connecting to central the cloud. Due to the high bandwidth usage cost and the performance cost of the WAN, processing the traffic in the remote central cloud is typically far more expensive than that across edge clouds. Given the amount $u_s(t)$ of input traffic outsourced to the central cloud from each source edge cloud s, the total cloud outsourcing cost at time slot t can be computed by:

$$C_O(t) = \sum_{s \in \mathcal{M}} r_s u_s(t).$$

4) Traffic Routing Cost: When routing traffic of various SFCs across multiple edge clouds, two kinds of different cost would be incurred by the cross-edge wide-area network (WAN) links. The first is the usage of the scarce and expensive WAN bandwidth. The second is the performance penalty incurred by the network latency of the cross-edge WAN links. Since both of these two kinds of cost are determined by the source edge cloud and destination edge cloud of the traffic routing process, we use a unified cost parameter d_{ij} to denote the overall cost (i.e., WAN bandwidth cost plus performance penalty) of routing one unit traffic from edge cloud i to edge cloud j.

For the traffic of SFC s processed by the cross-edge clouds, the total routing cost can be computed by summing up the followings: (1) the cost of routing the input traffic from the source edge cloud s to the first SF of the SFC s,[2]

[1] Here we focus on the static server power [26], since it contributes the majority of the whole server power consumption and overwhelms the dynamical server power. Nevertheless, since the static server power is proportional to the amount of traffic processed, it can be readily incorporated into the traffic routing cost which will be formulated later.

[2] Here we use a dummy SF 0 to represent the input process at each source edge cloud s

$\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{I}} h^s_{0m} z^m_{si}(t) d_{si}$. Here we perform summation over all the edge clouds $\mathcal{I}$, the rationale is that the instances of the first SF of each SFC s can be placed at multiples edge clouds for the purpose of service locality. (2) The cost of routing the intermediate traffic in each direct hop $m \rightarrow n$ of the SFC s ($h^s_{mn} = 1$), $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} h^s_{mn} y^{mn}_{sij}(t) d_{ij}$. (3) The routing cost of routing the output traffic from the edge clouds that host instances of the tail SF of SFC s back to the source edge cloud s ($h^s_{m0} = 1$), $\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{I}} h^s_{m0}\alpha^s_m z^m_{si}(t) d_{is}$. Summing up the above terms, the total cost of routing traffic of SFC s can be given by: $\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{I}} [h^s_{0m} z^m_{si}(t) d_{si} + h^s_{m0}\alpha^s_m z^m_{si}(t) d_{is}] + \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{M}}\sum_{j\in\mathcal{J}} h^s_{mn} y^{mn}_{sij}(t) d_{ij}$. Summing over all the SFC s, the overall routing cost $C_R(t)$ of all the traffic in time slot t is given by $\sum_s\sum_m\sum_i h^s_{0m} z^m_{si}(t) d_{si} + \sum_s\sum_m\sum_i h^s_{m0}\alpha^s_m z^m_{si}(t) d_{is} + \sum_s\sum_m\sum_i\sum_n\sum_j h^s_{mn} y^{mn}_{sij}(t) d_{ij}$. For ease of presentation, we let $g^m_{si} = h^s_{0m} d_{si} + h^s_{m0}\alpha^s_m d_{is}$, then the above term can be simplified to:

$$C_R(t) = \sum_{s\in\mathcal{S}}\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{I}}\Big(g^m_{si} z^m_{si}(t) + \sum_{n\in\mathcal{M}}\sum_{j\in\mathcal{I}} h^s_{mn} d_{ij} y^{mn}_{sij}(t)\Big).$$

D. The Cost Minimization Problem

In this paper, we aim to develop a cost-efficient service function chaining framework to facilitate the cross-edge deployment of SFCs, towards the goal of minimizing the holistic cost of the cross-edge system. To this end, we formulate a joint optimization on instance provisioning and traffic routing, aiming at minimizing the overall cost over the long-term.

$$\begin{aligned}
&\mathsf{P}:\\
&\min \sum_{t\in\mathcal{T}} [C_I(t) + C_S(t) + C_R(t) + C_O(t)],\\
&\text{s.t.} \sum_{s\in\mathcal{S}} z^m_{si}(t) \le x^m_i(t) b^m_i, \quad \forall t\in\mathcal{T},\ i\in\mathcal{I},\ m\in\mathcal{M}, && \text{(1a)}\\
&\quad \sum_{i\in\mathcal{I}} z^m_{si}(t) \ge \beta^m_s a^s_k(t), \quad \forall t\in\mathcal{T},\ s\in\mathcal{I},\ m\in\mathcal{M}, && \text{(1b)}\\
&\quad z^m_{si}(t) \ge \sum_{j\in\mathcal{I}}\sum_{n\in\mathcal{M}} h^s_{nm} y^{nm}_{sji}(t),\\
&\qquad \forall t\in\mathcal{T}, i,s\in\mathcal{I}, m\in\mathcal{M}/\{m^s_F\}, && \text{(1c)}\\
&\quad \alpha^m_s z^m_{si}(t) \le \sum_{j\in\mathcal{I}}\sum_{n\in\mathcal{M}} h^s_{mn} y^{mn}_{sij}(t),\\
&\qquad \forall t\in\mathcal{T}, i,s\in\mathcal{I}, m\in\mathcal{M}/\{m^s_L\}, && \text{(1d)}\\
&\quad a_s(t) + u_s(t) \ge A_s(t), \quad \forall t\in\mathcal{T},\ s\in\mathcal{I}, && \text{(1e)}\\
&\quad y^{mn}_{sij}(t) \ge 0, \quad \forall t\in\mathcal{T},\ i,j,s\in\mathcal{I},\ m,n\in\mathcal{M}, && \text{(1f)}\\
&\quad z^m_{si}(t) \ge 0, \quad \forall t\in\mathcal{T},\ i,s\in\mathcal{I},\ m\in\mathcal{M}, && \text{(1g)}\\
&\quad a_s(t) \ge 0, \quad \forall t\in\mathcal{T},\ s\in\mathcal{I}, && \text{(1h)}\\
&\quad u_s(t) \ge 0, \quad \forall t\in\mathcal{T},\ s\in\mathcal{I}, && \text{(1i)}\\
&\quad x^m_i(t) \in \{0, 1, \ldots, C^m_i\}, \quad \forall t\in\mathcal{T}, i\in\mathcal{I},\ m\in\mathcal{M}. && \text{(1j)}
\end{aligned}$$

Here m^s_F and m^s_L represent the head SF and tail SF of SFC s, respectively. Eq. (1a) is the capacity constraint which enforces that for each SF m, the traffic routed to each edge cloud i does not exceed the provisioned processing capacity $x^m_i(t) b^m_i$. Eq. (1b) is the load balancing constraint which indicates that the total incoming traffic of SFC s to SF m in all edge clouds should be no less than the aggregated traffic of SFC s at this SF m. Eq. (1c) and Eq. (1d) are the traffic reservation constraints that guarantee that for each SF in each edge cloud i, the outgoing traffic rate is no less than α^m_s times the incoming traffic rate. Eq. (1e) ensures that all the incoming input traffic can be served by either the edge clouds or the remote central cloud. Eq. (1f) – Eq. (1i) are the non-negative constraints for the decision variables. Finally, Eq. (1j) is the integrality constraint for the number of deployed SF instances.

In the above problem formulation P, we only consider the computing resource constraint and omit the network constraint. The rationale is that, typically edge servers are interconnected by high-speed local area network and computing resource sharing can be much more demanding [1], [6]. Also, in practice, the network bottleneck can be the cross-edge link, or uplink/downlink of each edge node, or both. However, to the best of our knowledge, there is no empirical measurement study identifying the network bottleneck in collaborative edge computing environments. To avoid misleading assumptions, we do not consider the constraint of network bandwidth capacity in this paper. We hope that our problem formulation would stimulate the research community to conduct empirical measurements to uncover the network bottleneck in collaborative edge computing environments, and we are glad to extend our model to incorporate this new constraint in our future work.

Solving the above optimization problem P is non-trivial due to the following dual challenges. First, the long-term cost minimization problem P is a time-coupling problem that involves further system information, as the instance switching cost $C_S(t)$ couples the decision of consecutive time slots. However, in realistic cross-edge system, parameters such as traffic arrival rates typically fluctuate over time and thus cannot be readily predicted. Then, how can we minimize the long-term cost in an online manner, without knowing the future information as a priori knowledge? Second, even with an offline setting where all the future information is given as a priori knowledge, the corresponding cost minimization problem is NP-hard. Specifically, our problem can be reduced from the classical minimum knapsack problem (MKP) [27] which is known to be NP-hard. Due to the space limit, interested readers are referred to the Appendix A of our online technical report [28] for the detailed reduction from the MKP.

IV. Online Optimization for Long-Term Cost Minimization

To address the dual challenges of time-coupling effect and NP-hardness of the long-term cost minimization problem P, we blend the advantages of a regularization method for online algorithm design and a dependent rounding technique for approximation algorithm design to propose a provably efficient online algorithm. The key idea of the proposed online algorithm is two-fold: (1) by regularizing the time-coupling switching cost $C_S(t)$ and relaxing the integer variable $x^m_i(t)$, we decompose the long-term problem into a series of one-shot fractional problems that can be readily solved. (2) By rounding the fractional solution with a dependent rounding scheme, we obtain a near-optimal solution to the original problem, with bounded optimality gap.

A. Problem Decomposition via Relaxation and Regularization

In response to the challenge of the NP-hardness of problem P, we first relax the integrality constraint Eq. (1j), obtaining the fractional optimization problem P_{R} as follows:

$$\begin{aligned} \mathsf{P}_{\mathsf{R}} :\min \ & \sum_{t\in\mathcal{T}} C_I(t) + C_S(t) + C_O(t) + C_R(t), \\ \text{s.t. } & \text{Constraint (1f) to (1i)}, \\ & x_i^m(t) \in [0, C_i^m], \quad \forall t \in \mathcal{T},\ i \in \mathcal{I},\ m \in \mathcal{M}. \end{aligned}$$

For the above fractional problem P_{R}, the relaxed switching cost $C_S(t)$ still temporally couples $x_i^m(t)$ across the time span. To address this issue, a natural solution would be greedily adopting the best decision for the relaxed problem in each independent time slot. However, this naive solution does not necessarily reach the global optimum for the long-term, and may even lead to arbitrary bad results.

Towards worst-case performance guarantee for the online algorithm of our problem, we exploit the algorithmic technique of regularization in online learning [29]. The basic idea of regularization is to solve the relaxed problem P_{R} with regularized objective function to substitute the intractable $[x_i^m(t) - x_i^m(t-1)]^+$. Specifically, in this paper, to approximate the term $[x_i^m(t) - x_i^m(t-1)]^+$, [29], we employ the widely adopted convex regularizer relative entropy function [29] as follows:

$$\begin{aligned} \Delta(x_i^m(t) || x_i^m(t-1)) = x_i^m(t) \ln \frac{x_i^m(t)}{x_i^m(t-1)} \\ + x_i^m(t) - x_i^m(t-1). \qquad (2) \end{aligned}$$

Here we obtain the relative entropy function by summing the relative entropy term $x_i^m(t) \ln \frac{x_i^m(t)}{x_i^m(t-1)}$ and a linear term denoting the movement $x_i^m(t) - x_i^m(t-1)$. Due to the convexity of the above regularizer $\Delta(x_i^m(t)||x_i^m(t-1))$, it has been widely adopted to approximate optimization problems that involve L1-distance terms (e.g., the switching cost $C_S(t)$ in our problem) in online learning. To ensure that the fraction is still valid when no instance of SF m is deployed in edge cloud i at time slot $t-1$ (i.e., $x_i^m(t-1) = 0$), we add a positive constant term ϵ to both $x_i^m(t)$ and $x_i^m(t-1)$ in the relative entropy term in Eq. (2). To normalize the switching cost $C_S(t)$ by regularization, we also define an approximation weight factor $\eta_i^m = \ln(1 + \frac{C_i^m}{\epsilon})$ and multiply the improved relative entropy function by $\frac{1}{\eta_i^m}$.

By using the enhanced regularizer $\Delta(x_i^m(t)||x_i^m(t-1))$ to approximate the time-coupling term $[x_i^m(t) - x_i^m(t-1)]^+$ in the switching cost $C_S(t)$, we further obtain the relaxed and regularized problem which is denoted as P_{Re}. Though P_{Re} is still time-coupling, the convex, differentiable and logarithmic-based regularizer $\Delta(x_i^m(t)||x_i^m(t-1))$ enables us to temporally decouple P_{Re} into a series of one-shot convex programs $\mathsf{P}_{\mathsf{Re}}^{\mathsf{t}}$, which can be solved in each individual time slot t based the solution obtained from the previous time slot $t-1$. These series of solutions generated in each time slot thus constitute a feasible yet near-optimal solution to our original problem, with provable performance guarantee even for the worst-case (to be analyzed in Sec. V). Specifically, the decomposed subproblem $\mathsf{P}_{\mathsf{Re}}^{\mathsf{t}}$ for each time slot t, $\forall t \in \mathcal{T}$ can be denoted as follows:

$$\begin{aligned} \mathsf{P}_{\mathsf{Re}}^{\mathsf{t}} :\min \ & C_I(t) + C_O(t) + C_R(t) + \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} \frac{q_i^m}{\eta_i^m} \Bigg(\big(x_i^m(t) \\ & + \epsilon\big) \ln \frac{x_i^m(t) + \epsilon}{x_i^m(t-1) + \epsilon} + x_i^m(t-1) - x_i^m(t) \Bigg), \\ \text{s.t. } & \text{Constraint (1f) to (1i)}, \\ & x_i^m(t) \in [0, C_i^m], \quad \forall t \in \mathcal{T},\ i \in \mathcal{I},\ m \in \mathcal{M}. \end{aligned}$$

Since the problem $\mathsf{P}_{\mathsf{Re}}^{\mathsf{t}}$ is a standard convex optimization with linear constraints, it can be optimally solved in polynomial time, by taking existing convex optimization technique as exemplified by the classical interior point method [30].

Note that at each time slot t, the variable $x_i^m(t-1)$, $\forall i \in \mathcal{I}, m \in \mathcal{M}$ has been obtained when solving $\mathsf{P}_{\mathsf{Re}}^{\mathsf{t-1}}$ at time slot $t-1$, and it is required as the input to solve $\mathsf{P}_{\mathsf{Re}}^{\mathsf{t}}$ at time slot t. In this regard, we develop an Online Regularization-based Fractional Algorithm (ORFA) as shown in Alg. 1, which generates an optimal fractional solution $(\widetilde{\mathbf{x}}(t), \widetilde{\mathbf{y}}(t), \widetilde{\mathbf{z}}(t), \widetilde{\mathbf{a}}(t), \widetilde{\mathbf{u}}(t))$ at each time slot t by using the previous and current system information. It is obvious that this optimal solution of the relaxed and regularized problem $\mathsf{P}_{\mathsf{Re}}^{\mathsf{t}}$, constitutes a feasible solution to the relaxed (but unregularized) problem P_{R}. Later, this feasibility will be leveraged to derive the competitive ratio of the ORFA algorithm in Sec. V.

Algorithm 1 An Online Regularization-Based Fractional Algorithm — ORFA

```
Input: $\mathcal{I}, \mathcal{M}, \mathcal{S}, \boldsymbol{b}, \boldsymbol{C}, \boldsymbol{g}, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{q}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \epsilon$
Output: $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{a}$
Initialization: $\boldsymbol{x} = \mathbf{0}, \boldsymbol{y} = \mathbf{0}, \boldsymbol{z} = \mathbf{0}, \boldsymbol{u} = \mathbf{0}, \boldsymbol{a} = \mathbf{0}$;
for each time slot $t \in \mathcal{T}$ do
    Observe values of $\boldsymbol{A}(t), \boldsymbol{q}(t)$ and $\boldsymbol{x}(t-1)$;
    Invoke the interior point method to solve the regularized
        problem $\mathsf{P}_{\mathsf{Re}}^{\mathsf{t}}$;
    return the optimal fractional solution $\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{z}(t)$;
end for
```

B. A Randomized Dependent Rounding Scheme

The proposed online regularization-based fractional algorithm ORFA obtains a fractional solution of problem $\mathsf{P}_{\mathsf{Re}}^{\mathsf{t}}$. In order to satisfy the integrality constraint Eq. (1j) of the original problem P, we need to round the optimal fractional solution $\widetilde{\boldsymbol{x}}(t)$ to an integer solution $\bar{\boldsymbol{x}}(t)$. To this end, a straightforward solution is the independent randomized rounding scheme [31], whose basic idea is to round up each fractional $\widetilde{\boldsymbol{x}}(t)$ to the nearest integer $\bar{\boldsymbol{x}}(t) = \lceil \widetilde{\boldsymbol{x}}(t) \rceil$ with a probability of $\widetilde{\boldsymbol{x}}(t) - \lfloor \widetilde{\boldsymbol{x}}(t) \rfloor$, i.e., $\Pr\{\bar{\boldsymbol{x}}(t) = \lceil \widetilde{\boldsymbol{x}}(t) \rceil\} = \widetilde{\boldsymbol{x}}(t) - \lfloor \widetilde{\boldsymbol{x}}(t) \rfloor$, and round down $\widetilde{\boldsymbol{x}}(t)$ to the nearest integer $\bar{\boldsymbol{x}}(t) = \lfloor \widetilde{\boldsymbol{x}}(t) \rfloor$ with a probability of $\lceil \widetilde{\boldsymbol{x}}(t) \rceil - \widetilde{\boldsymbol{x}}(t)$, i.e., $\Pr\{\bar{\boldsymbol{x}}(t) = \lfloor \widetilde{\boldsymbol{x}}(t) \rfloor\} = \lceil \widetilde{\boldsymbol{x}}(t) \rceil - \widetilde{\boldsymbol{x}}(t)$.

While the above independent rounding policy can always generate a feasible solution (since the remote central cloud is able to cover all the unserved traffic incurred by rounding down $\widetilde{x}_i^m(t)$), directly applying this policy to round $\widetilde{x}_i^m(t)$ may incur high instance running cost or cloud outsourcing cost. That is, with certain probability, an excessive amount or even all the SF instances at each edge cloud are destroyed, leading to the situation that an enormous amount of input traffic are outsourced to the expensive central cloud. Similarly, an excessive amount of SF instances may also be launched due to aggressively rounding up $\widetilde{x}_i^m(t)$), leading to a sharp growth of the instance running cost.

To address the above challenge, we therefore develop a randomized and dependent pairwise rounding scheme [10] that can exploit the inherent dependence of the variables $\widetilde{x}_i^m(t)$. The key idea is that, a rounded-down variable will be compensated by another rounded-up variable, ensuring that the input traffic absorbed by the edge clouds could be fully processed by the edge clouds even after the rounding phase. With such a dependent rounding scheme, the variables would not be aggressively rounded up or down, reducing the cost of using the expensive central cloud or launching an excessive amount of SF instances at the edge clouds.

For each SF $m \in \mathcal{M}$, we introduce two sets $\mathcal{I}_{mt}^+ = \{i|\widetilde{x}_i^m(t) \in \mathbb{Z}\}$ and $\mathcal{I}_{mt}^- = \{i|\widetilde{x}_i^m(t) \in \mathbb{R}^+\}$. Intuitively, $\mathcal{I}_{mt}^+$ denotes the set of edge clouds with integral $\widetilde{x}_i^m(t)$ while $\mathcal{I}_{mt}^-$ denotes the set of edge clouds with fractional $\widetilde{x}_i^m(t)$ and thus should be rounded. According to the above definitions, we have $\mathcal{I}_{mt}^+ \bigcup \mathcal{I}_{mt}^- = \mathcal{I}$ at each time slot t. For each element $i \in \mathcal{I}_{mt}^-$, we further introduce a probability coefficient p_i^m and a weight coefficient ω_i^m associated with it. Here we define $p_i^m = \widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor$, and $\omega_i^m = b_i^m, \forall m \in \mathcal{M}, i \ \in \mathcal{I}_{mt}^-$.

The detailed Randomized Dependent Instance Provisioning (RDIP) algorithm which rounds the fractional solution is shown in Alg. 2. Specifically, for each SF $m \in \mathcal{M}$, to round the elements in $\mathcal{I}_{jt}^-$ with fractional $x_{ij}(t)$, the proposed RDIP algorithm runs a series of rounding iterations. At each iteration, we randomly select two elements i_1 and i_2 from $\mathcal{I}_{mt}^-$, and let the probability of one of these two elements round to 0 or 1, decided by the coupled coefficient γ_1 and γ_2. By doing so, at each iteration, the number of elements in $\mathcal{I}_{mt}^-$ would decrease at least by 1. Finally, when $\mathcal{I}_{mt}^-$ has only one element in the last iteration, we directly round it up with its current probability.

The proposed rounding scheme is cost-efficient, in terms of that it would not aggressively launch new SF instances or outsource unserved input traffic to the central cloud. This cost-efficiency is achieved by maintaining three desirable properties in the main loop of each iteration.

Firstly, **continuous reduction property**. At least one of the two selected variables $\widetilde{x}_{i_1}^m(t)$ and $\widetilde{x}_{i_2}^m(t)$ is rounded into integer. For example, if $\varphi_1 = 1 - p_{i_1}^m$ and $\varphi_2 = p_{i_1}^m$ (line 11), then $p_{i_1}^m = 1$ if line 13 is executed and $p_{i_1}^m = 0$ if line 15 is executed. In both cases, $\widetilde{x}_{i_1}^m(t)$ will be rounded to a integer.

Secondly, **weight conservation property**. That is, after the main loop of each iteration, the total weighted resource capacity of the selected two elements (i.e., SF instance) keeps unchange, i.e., the sum $\widetilde{x}_{i_1}^m(t)b_{i_1}^m + \widetilde{x}_{i_2}^m(t)b_{i_2}^m$ stays constant. For example, if line 13 is executed, we have $[\widetilde{x}_{i_1}^m(t)+\varphi_1]b_{i_1}^m + [\widetilde{x}_{i_2}^m(t) - \frac{b_{i_1}^m}{b_{i_2}^m}\varphi_1]b_{i_2}^m = \widetilde{x}_{i_1}^m(t)b_{i_1}^m + \widetilde{x}_{i_2}^m(t)b_{i_2}^m$. Similarly, if line 15 is executed, we can also prove that this equation still holds.

Algorithm 2 Randomized Dependent Instance Provision — RDIP

```
Input: $\mathcal{I}, \mathcal{M}, \widetilde{\boldsymbol{x}}(t-1), \boldsymbol{b}$
Output: $\bar{\boldsymbol{x}}(t)$
1: for each SF $m \in \mathcal{M}$ do
2:   Let $\mathcal{I}_{mt}^+ = \{i|\widetilde{x}_i^m(t) \in \mathbb{Z}\}, \mathcal{I}_{mt}^- = \{i|\widetilde{x}_i^m(t) \in \mathbb{R}^+\}$;
3:   for each edge cloud $i \in \mathcal{I}_{it}^+$ do
4:     Set $\bar{x}_i^m(t) = \widetilde{x}_i^m(t)$;
5:   end for
6:   for each edge cloud $i \in \mathcal{I}_{it}^-$ do
7:     Let $p_i^m = \widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor, \omega_i^m = b_i^m$;
8:   end for
9:   while $|\mathcal{I}_{mt}^-| > 1$ do
10:    Randomly select two elements $i_1, i_2$ from $\mathcal{I}_{mt}^-$;
11:    Define $\varphi_1 = \min\{1 - p_{i_1}^m, \frac{\omega_{i_2}^m}{\omega_{i_1}^m} p_{i_2}^m\}$, $\varphi_2 = \min\{p_{i_1}^m, \frac{\omega_{i_2}^m}{\omega_{i_1}^m}(1 - p_{i_2}^m)\}$;
12:    With the probability $\frac{\varphi_2}{\varphi_1+\varphi_2}$ set
13:    $p_{i_1}^m = p_{i_1}^m + \varphi_1, p_{i_2}^m = p_{i_2}^m - \frac{\omega_{i_1 j}}{\omega_{i_2 j}}\varphi_1$;
14:    With the probability $\frac{\varphi_1}{\varphi_1+\varphi_2}$ set
15:    $p_{i_1}^m = p_{i_1}^m - \varphi_2, p_{i_2}^m = p_{i_2}^m + \frac{\omega_{i_1}^m}{\omega_{i_2}^m}\varphi_2$;
16:    If $p_{i_1}^m \in \{0,1\}$, then set $\bar{x}_{i_1}^m(t) = \lfloor \widetilde{x}_{i_1}^m(t) \rfloor + p_{i_1}^m$,
17:    $\mathcal{I}_{mt}^+ = \mathcal{I}_{mt}^+ \cup \{i_1\}, \mathcal{I}_{mt}^- = \mathcal{I}_{mt}^- \setminus \{i_1\}$;
18:    If $p_{i_2}^m \in \{0,1\}$, then set $\bar{x}_{i_2}^m(t) = \lfloor \widetilde{x}_{i_2}^m(t) \rfloor + p_{i_2}^m$,
19:    $\mathcal{I}_{mt}^+ = \mathcal{I}_{mt}^+ \cup \{i_2\}, \mathcal{I}_{mt}^- = \mathcal{I}_{mt}^- \setminus \{i_2\}$;
20:  end while
21:  if $|\mathcal{I}_{mt}^-| = 1$ then
22:    Set $\bar{x}_i^m(t) = \lceil \widetilde{x}_i^m(t) \rceil$ for the only element $i \in \mathcal{I}_{mt}^-$;
23:  end if
24: end for
```

Thirdly, **marginal distribution property**. That is, the probability of rounding up or down each element $i \in \mathcal{I}_{mt}^-, \forall m \in \mathcal{M}$ after the main loop is determined by the fractional part $\widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor$ of the fractional solution $\widetilde{x}_i^m(t)$. More specifically, $\Pr\{\bar{x}_i^m(t) = \lceil \widetilde{x}_i^m(t) \rceil\} = \widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor, \Pr\{\bar{x}_i^m(t) = \lfloor \widetilde{x}_i^m(t) \rfloor\} = 1 - (\widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor)$. Based on this marginal distribution property which has been proven in [10], we can further derive:

$$\begin{aligned}\mathbb{E}\{\bar{x}_i^m(t)\} =& (\widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor)\lceil \widetilde{x}_i^m(t) \rceil \\ &+ [1 - (\widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor)]\lfloor \widetilde{x}_i^m(t) \rfloor \\ =& (\widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor)(\lfloor \widetilde{x}_i^m(t) \rfloor + 1) \\ &+ [1 - (\widetilde{x}_i^m(t) - \lfloor \widetilde{x}_i^m(t) \rfloor)]\lfloor \widetilde{x}_i^m(t) \rfloor \\ =& \widetilde{x}_i^m(t).\end{aligned}$$

The above equation shows that the expectation of each rounded solution is exactly the value of the original fractional solution. This property indicates that new SF instances would not be aggressively launched when rounding the fractional solution. As a result, the optimality gap between the expected total cost incurred by the rounded solution and that incurred by the optimal fractional solution is bounded, as we will prove in Sec. V-C.

C. Traffic Re-Routing

After performing the randomized and dependent rounding scheme RDIP at each time slot t, the instance provisioning decision $\bar{\boldsymbol{x}}(t)$ produced by RDIP algorithm together with the traffic routing decision $(\widetilde{\boldsymbol{y}}(t), \widetilde{\boldsymbol{z}}(t), \widetilde{\boldsymbol{u}}(t), \widetilde{\boldsymbol{a}}(t))$ may well not be a feasible solution of the original problem P. This means that we further need to modify the fractional traffic routing decision $(\widetilde{\boldsymbol{y}}(t), \widetilde{\boldsymbol{z}}(t), \widetilde{\boldsymbol{u}}(t), \widetilde{\boldsymbol{a}}(t))$ accordingly, in order to maintain the solution feasibility.

To the above end, a naive approach is to bring the feasible integral solution $\bar{\boldsymbol{x}}(t)$ back to the original problem P, and solve the degraded traffic re-routing problem to obtain a complete feasible solution $(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{y}}(t), \bar{\boldsymbol{z}}(t), \bar{\boldsymbol{u}}(t), \bar{\boldsymbol{a}}(t))$. However, if we further consider the weight conservation property maintained by the randomized dependent rounding scheme in Sec. IV-B, we may obtain a complete feasible solution in a simpler manner. Specifically, the weight conservation property of the RDIP algorithm ensures that for each SF $m \in \mathcal{M}$, the total resource capacity $\sum_i b_i^m \bar{x}_i^m$ of the rounded solution is no smaller than $\sum_i b_i^m \widetilde{x}_i^m$ of the fractional solution. This further indicates that the resource capacity of the rounded solution is sufficient to cover the total traffic absorbed by the edge clouds when performing the ORFA algorithm. Then, an intuition is to let $\bar{\boldsymbol{u}}(t) = \widetilde{u}(t)$, $\bar{\boldsymbol{a}}(t)) = \widetilde{\boldsymbol{a}}(t)$ and solve the following simplified traffic re-routing problem to obtain $\bar{\boldsymbol{y}}(t)$ and $\bar{\boldsymbol{z}}(t)$.

$$
\begin{aligned}
&\min \ C_R(t) \\
&\text{s.t. Constraints (1a)-(1d), (1f)} \quad \text{and (1g).}
\end{aligned}
$$

Since this traffic re-routing problem is a linear programming, therefore its optimal solution can be readily computed in polynomial time, by applying linear programming techniques as exemplified by interior point method.

Remark: in terms of optimality, we acknowledge that the weight conservation property based traffic re-routing scheme is not as better as the aforementioned naive approach, since the latter has a larger feasible space. However, in a realistic cross-edge system, the cost of cloud outsourcing is far more expensive than edge-based processing, the optimality gap between those two approaches is small actually. Besides, here we study the former one due to the following three reasons: (1) it has lower time complexity; (2) it is more straightforward for performance analysis (to be detailed in Sec. V); and (3) it can work as a bridge to analyze the performance bound of the aforementioned naive approach for traffic re-routing, as we will discuss later in Sec. V-C.

V. Performance Analysis

In this section, we rigorously analyze the theoretical performance of the proposed online algorithm via competitive analysis. As a standard approach to quantifying the performance of online algorithms, the basic idea of competitive analysis is to compare the performance of the online algorithm to the theoretical optimal solution in the offline case, where all the future information is given as a priori knowledge. In particular, we prove that the proposed online algorithm has guaranteed performance, which is quantified by a worst-case parameterized competitive ratio.

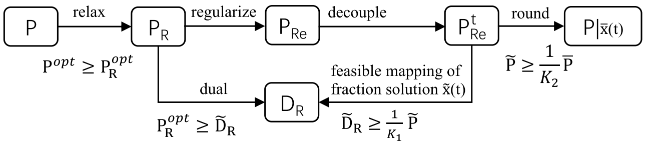
P
relax
P_R
regularize
P_Re
decouple
P^t_Re
round
$\mathsf{P}|\bar{x}(t)$
$\mathsf{P}^{opt} \geq \mathsf{P}^{opt}_\mathsf{R}$
dual
D_R
feasible mapping of
fraction solution $\tilde{x}(t)$
$\widetilde{\mathsf{P}} \geq \frac{1}{K_2}\bar{\mathsf{P}}$
$\mathsf{P}^{opt}_\mathsf{R} \geq \widetilde{\mathsf{D}}_\mathsf{R}$
$\widetilde{\mathsf{D}}_\mathsf{R} \geq \frac{1}{K_1}\widetilde{\mathsf{P}}$

Fig. 3. An illustration of the basic idea of the performance analysis.

A. The Basic Idea

Essentially, we derive the competitive ratio based on the well-established primal-dual framework for convex optimization. To this end, as illustrated in Fig. 3, we introduce the dual problem of the relaxed problem D_R to act as the bridge that connects the original problem and regularized problem through the following chain of inequalities.

$$
\mathsf{P}^{opt} \geq \mathsf{P}^{opt}_\mathsf{R} \geq \widetilde{\mathsf{D}}_\mathsf{R} \geq \frac{1}{K_1}\widetilde{\mathsf{P}} \geq \frac{1}{K_1\ K_2}\bar{\mathsf{P}}. \tag{4}
$$

Here P^{opt} and $\mathsf{P}^{opt}_\mathsf{R}$ denote the objective values achieved by the optimal solutions of the original problem P and relaxed problem P_R, respectively. $\widetilde{\mathsf{D}}_\mathsf{R}$ denotes the objective value of P_R's dual problem D_R, achieved by a feasible solution mapped from the optimal fractional solution of the regularized problem P_Re. Finally, $\widetilde{\mathsf{P}}$ and $\bar{\mathsf{P}}$ are the objective values of the original problem P, achieved by the optimal fractional solution and the rounded solution, respectively.

Since P_R is a relaxed problem of the original minimization problem P, we thus have $\mathsf{P}^{opt} \geq \mathsf{P}^{opt}_\mathsf{R}$. The inequality $\mathsf{P}^{opt}_\mathsf{R} \geq$ D_R holds due to the celebrated Weak Duality Theorem in convex optimization theory. Besides, to connect the dual problem D_R to the regularized problem P_Re, we construct a feasible solution for D_R mapped from the optimal fractional solution for P_Re. Based on such mapping, we can derive the competitive ratio K_1 by applying Karush-Kuhn-Tucker (KKT) conditions, i.e., the first-order necessary conditions characterizing the optimal solution to exploit the inherent common structures shared by P_Re and D_R. Finally, based on the three desirable properties maintained by the dependent rounding scheme, we can further characterize the competitive ratio K_2 between the optimal fractional solution and the rounded solution.

B. Competitive Ratio of ORFA

We establish $\widetilde{\mathsf{D}}_\mathsf{R} \geq \frac{1}{K_1}\widetilde{\mathsf{P}}$ and derive the competitive ration K_1 of ORFA in this subsection. Specifically, we first drive the Lagrange dual problem D_R of the relaxed problem P_R. Then, we characterize the the optimality conditions of the regularized problem P_Re. Finally, we construct a feasible solution for D_R mapped from the primal and dual optimal solutions for P_Re.

1) An Equivalent Problem Transformation: Directly writing the Lagrange dual problem D_R for the relaxed cost minimization problem P_R is not straightforward, due to the time-coupling switching cost $C_S(t)$ and the boxing constraints of control variables $x_i^m(t)$. In response, we first introduce a set of new variables $w_i^m(t)$ which satisfy $w_i^m(t) \geq x_i^m(t) - x_i^m(t-1), \forall i \in \mathcal{I}, m \in \mathcal{M}$ to replace the time-coupling term $[x_i^m(t) - x_i^m(t-1)]^+$. Note that since we further enforce that $w_i^m(t) \geq 0, \forall i \in \mathcal{I}, m \in \mathcal{M}$, the transformed switching cost $C_s(t) = \sum_i \sum_m q_i^m w_i^m(t)$ is equivalent to the original expression. For the boxing constraints $x_i^m(t) \in \{0, 1, 2, \ldots, C_i^m\}$ (Eq. (1j)), as suggested by the literature [29], we replace it by a set of knapsack cover (KC) constraints $\sum_{j\in\mathcal{I}} x_j^m(t) b_j^m - x_i^m(t) b_i^m \geq \sum_{s\in\mathcal{I}} \beta_s^m [A_s(t) - u_s(t)] - C_i^m b_i^m$. For ease of representation, we denote $B_i^m(t) = \sum_{s\in\mathcal{I}} \beta_s^m A_s(t) - C_i^m b_i^m$ hereafter. With the above transformations, we rewrite the original cost minimization problem in the following equivalent form.

$$\min \sum_{t\in\mathcal{T}} [C_I(t) + \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} q_i^m w_i^m(t) + C_R(t) + C_O(t)],$$

$$\text{s.t. } w_i^m(t) \geq x_i^m(t) - x_i^m(t-1), \quad \forall t \in \mathcal{T}, \quad \forall i \in \mathcal{I}, m \in \mathcal{M}, \tag{5a}$$

$$\sum_{s\in\mathcal{S}} z_{si}^m(t) \leq x_i^m(t) b_i^m, \quad \forall t \in \mathcal{T},\ i \in \mathcal{I},\ m \in \mathcal{M}, \tag{5b}$$

$$\sum_{i\in\mathcal{I}} z_{si}^m(t) \geq \beta_s^m a_s(t),\ \forall t \in \mathcal{T},\ s \in \mathcal{I},\ m \in \mathcal{M}, \tag{5c}$$

$$z_{si}^m(t) \geq \sum_{j\in\mathcal{I}} \sum_{n\in\mathcal{M}} h_{nm}^s y_{sji}^{nm}(t), \quad \forall t \in \mathcal{T}, i,\ s \in \mathcal{I},\ m \in \mathcal{M}/\{m_F^s\}, \tag{5d}$$

$$\alpha_s^m z_{si}^m(t) \leq \sum_{j\in\mathcal{I}} \sum_{n\in\mathcal{M}} h_{mn}^s y_{sij}^{mn}(t), \quad \forall t \in \mathcal{T}, i,\ s \in \mathcal{I},\ m \in \mathcal{M}/\{m_L^s\}, \tag{5e}$$

$$a_s(t) + u_s(t) \geq A_s(t), \quad \forall t \in \mathcal{T},\ s \in \mathcal{I}, \tag{5f}$$

$$\sum_{j\in\mathcal{I}} x_j^m(t) b_j^m - x_i^m(t) b_i^m \geq B_i^m(t) - \sum_{s\in\mathcal{I}} \beta_s^m u_s(t), \quad \forall t \in \mathcal{T}, i \in \mathcal{I}, m \in \mathcal{M}, \tag{5g}$$

$$y_{sij}^{mn}(t) \geq 0, \quad \forall t \in \mathcal{T}, i, j,\ s \in \mathcal{I},\ m, n \in \mathcal{M}, \tag{5h}$$

$$z_{si}^m(t) \geq 0, \quad \forall t \in \mathcal{T}, i,\ s \in \mathcal{I},\ m \in \mathcal{M}, \tag{5i}$$

$$a_s(t), u_s(t) \geq 0, \quad \forall t \in \mathcal{T},\ s \in \mathcal{I}, \tag{5j}$$

$$x_i^m(t), w_i^m(t) \geq 0, \quad \forall t \in \mathcal{T},\ i \in \mathcal{I},\ m \in \mathcal{M}. \tag{5k}$$

2) Deriving the Lagrange Dual Problem: We are now ready to derive the Lagrange dual problem $\mathsf{D_R}$ of the above transformed problem (and also equally the problem $\mathsf{P_R}$). Here we use $\nu_i^m(t)$, $\lambda_i^m(t)$, $\rho_s^m(t)$, $\gamma_{si}^m(t)$, $\tau_{si}^m(t)$, $\theta_s(t)$ and $\phi_i^m(t)$ to denote the corresponding dual variables for the constraints Eq. (5a) to Eq. (5g). Then the dual problem $\mathsf{D_R}$ can be written as follows.

$$\mathsf{D_R} : \min \sum_{t\in\mathcal{T}} \left\{ \sum_{s\in\mathcal{I}} A_s(t)\theta_s(t) + \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} B_i^m(t)\phi_i^m(t) \right\}$$

$$\text{s.t. } -\nu_i^m(t) + \nu_i^m(t+1) + b_i^m \lambda_i^m(t) \leq p_i^m(t) - b_i^m [\sum_{j\in\mathcal{I}} \phi_j^m(t) - \phi_i^m(t)], \tag{6a}$$

$$\nu_i^m(t) \leq q_i^m, \tag{6b}$$

$$\theta_s(t) + \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} \phi_i^m(t) \beta_i^m \leq r_s(t), \tag{6c}$$

$$\theta_s(t) - \sum_{m\in\mathcal{M}} \rho_s^m(t) \beta_s^m \leq 0, \tag{6d}$$

$$\rho_s^m(t) - \lambda_i^m(t) + \gamma_{si}^m(t) - \alpha_s^m \tau_{si}^m(t) \leq g_{si}^m(t), \tag{6e}$$

$$\rho_s^m(t) - \lambda_i^m(t) - \alpha_s^m \tau_{si}^m(t) \leq g_{si}^m(t), \tag{6f}$$

$$\rho_s^m(t) - \lambda_i^m(t) + \gamma_{si}^m(t) \leq g_{si}^m(t), \tag{6g}$$

$$h_s^{mn} \tau_{si}^m(t) - h_s^{mn} \gamma_{sj}^n(t) \leq h_s^{mn} d_{ij}. \tag{6h}$$

All the dual variables ≥ 0.

3) Characterizing the Optimality of the Regularized Problem: As we regularize the relaxed problem $\mathsf{P_R}$ to a solvable convex problem $\mathsf{P_{Re}}$, we know that the optimal fractional solution $(\widetilde{\mathbf{x}}(t), \widetilde{\mathbf{y}}(t), \widetilde{\mathbf{z}}(t), \widetilde{\mathbf{a}}(t), \widetilde{\mathbf{u}}(t))$ obtained by OFRA satisfies the Karush-Kuhn-Tucker (KKT) conditions, i.e., the first-order necessary conditions for optimality. Here we use $(\widetilde{\lambda}_i^m(t)$, $\widetilde{\rho}_s^m(t)$, $\widetilde{\gamma}_{si}^m(t)$, $\widetilde{\tau}_{si}^m(t)$, $\widetilde{\theta}_s(t)$, $\widetilde{\phi}_i^m(t))$ to denote the optimal solution to the dual problem of the regularized problem $\mathsf{P_{Re}}$, corresponding to constraints Eq. (5a) to Eq. (5g), respectively. Then, the KKT conditions of the optimal fractional solution can be readily obtained. For easy of presentation, we write the KKT conditions in the following disjunctive form, where $a \perp b$ is equivalent to $a, b \geq 0$ and $ab = 0$.

$$\widetilde{\lambda}_i^m(t) \perp \Big(\widetilde{x}_i^m(t) b_i^m - \sum_{s\in\mathcal{I}} \widetilde{z}_{si}^m(t) \Big), \quad \forall t, s, i, m, \tag{7a}$$

$$\widetilde{\rho}_s^m(t) \perp \Big(\sum_{i\in\mathcal{I}} \widetilde{z}_{si}^m(t) - \beta_s^m \widetilde{a}_s(t) \Big), \quad \forall t, s, m, \tag{7b}$$

$$\widetilde{\gamma}_{si}^m(t) \perp \Big(\widetilde{z}_{si}^m(t) - \sum_{j\in\mathcal{I}} \sum_{n\in\mathcal{M}} h_{nm}^s \widetilde{y}_{sji}^{nm}(t) \Big), \quad \forall t, s, i, m \neq m_F^s, \tag{7c}$$

$$\widetilde{\tau}_{si}^m(t) \perp \Big(\sum_{j\in\mathcal{I}} \sum_{n\in\mathcal{M}} h_{nm}^s \widetilde{y}_{sji}^{nm}(t) - \widetilde{z}_{si}^m(t) \Big), \quad \forall t, s, i, m \neq m_L^s, \tag{7d}$$

$$\widetilde{\theta}_s(t) \perp \Big(\widetilde{a}_s(t) + \widetilde{u}_s(t) - A_s(t) \Big), \quad \forall t, s, \tag{7e}$$

$$\widetilde{\phi}_i^m(t) \perp \Big(\sum_{j\in\mathcal{I}} \widetilde{x}_j^m(t) b_j^m - \widetilde{x}_i^m(t) b_i^m - \big(B_i^m(t) - \sum_{s\in\mathcal{I}} \beta_s^m \widetilde{u}_s(t) \big) \Big), \quad \forall t, i, m, \tag{7f}$$

$$\widetilde{x}_i^m(t) \perp \Big(p_i^m(t) + \frac{q_i^m}{\eta_i^m} \ln \frac{\widetilde{x}_i^m(t) + \epsilon}{\widetilde{x}_i^m(t-1) + \epsilon} - b_i^m \Big(\sum_{j\in\mathcal{I}} \widetilde{\phi}_j^m(t) - \widetilde{\phi}_i^m(t) \Big) - b_i^m \widetilde{\lambda}_i^m(t) \Big), \quad \forall t, i, m \tag{7g}$$

$$\widetilde{z}_{si}^m(t) \perp \Big(g_{si}^m + \widetilde{\lambda}_i^m(t) - \widetilde{\rho}_s^m(t) - \widetilde{\gamma}_{si}^m(t) + \alpha_s^m \widetilde{\tau}_{si}^m(t) \Big), \quad \forall t, s, i, m \notin \{m_F^s, m_L^s\}, \tag{7h}$$

$$\widetilde{z}_{si}^m(t) \perp \Big(g_{si}^m + \widetilde{\lambda}_i^m(t) - \widetilde{\rho}_s^m(t) + \alpha_s^m \widetilde{\tau}_{si}^m(t) \Big), \quad \forall t, s, i, m = m_F^s, \tag{7i}$$

$$\widetilde{z}_{si}^{m}(t)\perp\Big(g_{si}^{m}+\widetilde{\lambda}_{i}^{m}(t)-\widetilde{\rho}_{s}^{m}(t)-\widetilde{\gamma}_{si}^{m}(t)\Big),\quad \forall t,s,i,m=m_{L}^{s}, \tag{7j}$$

$$\widetilde{a}_{s}(t)\perp\Big(\sum_{m\in\mathcal{M}}\widetilde{\rho}_{s}^{m}(t)\beta_{s}^{m}-\widetilde{\theta}_{s}(t)\Big),\quad \forall t,s, \tag{7k}$$

$$\widetilde{u}_{s}(t)\perp\Big(r_{s}(t)-\sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}}\widetilde{\phi}_{i}^{m}(t)\beta_{s}^{m}-\widetilde{\theta}_{s}(t)\Big)=0,\quad \forall t,s, \tag{7l}$$

$$\widetilde{y}_{sij}^{mn}(t)\perp\Big(h_{s}^{mn}d_{ij}+h_{s}^{mn}\widetilde{\gamma}_{sj}^{n}(t)-h_{s}^{mn}\widetilde{\tau}_{si}^{m}(t)\Big),\quad \forall t,s,i,j,m,n. \tag{7m}$$

4) Constructing a Mapping From the Regularized Problem to the Relaxed Dual Problem: Having characterized the optimality conditions for the regularized problem P_{Re}, we now construct a mapping to jointly map P_{Re}'s optimal primal and dual solutions to a feasible solution of the relaxed dual problem D_{R}. Here we denote the constructed mapping by[3]

$$\begin{aligned}&\boldsymbol{\Omega}\{\widetilde{\boldsymbol{x}}(t),\widetilde{\boldsymbol{y}}(t),\widetilde{\boldsymbol{z}}(t),\widetilde{\boldsymbol{a}}(t),\widetilde{\boldsymbol{u}}(t)\}\\&\quad=(\nu_{i}^{m}(t),\lambda_{i}^{m}(t),\rho_{s}^{m}(t),\gamma_{si}^{m}(t),\tau_{si}^{m}(t),\theta_{s}(t),\phi_{i}^{m}(t)),\end{aligned}$$

in which we let

$$\begin{aligned}&\nu_{i}^{m}(t)=\frac{q_{i}^{m}}{\eta_{i}^{m}}\ln\frac{C_{i}^{m}(t)+\epsilon}{\widetilde{x}_{i}^{m}(t-1)+\epsilon},\quad \lambda_{i}^{m}(t)=\widetilde{\lambda}_{i}^{m}(t),\\&\rho_{s}^{m}(t)=\widetilde{\rho}_{s}^{m}(t),\\&\gamma_{si}^{m}(t)=\widetilde{\gamma}_{si}^{m}(t),\quad \tau_{si}^{m}(t)=\widetilde{\tau}_{si}^{m}(t),\\&\theta_{s}(t)=\widetilde{\theta}_{s}(t),\phi_{i}^{m}(t)=\widetilde{\phi}_{i}^{m}(t).\end{aligned}$$

Lemma 1: The constructed mapping $\boldsymbol{\Omega}\{\widetilde{\boldsymbol{x}}(t),\widetilde{\boldsymbol{y}}(t),\widetilde{\boldsymbol{z}}(t),\widetilde{\boldsymbol{a}}(t),\widetilde{\boldsymbol{u}}(t)\}$ *is a feasible solution of the relaxed dual problem* D_{R}.

In Appendix. B of our online technical report [28], we prove this lemma by showing that the constraints Eq. (6a) to Eq. (6h) of the relaxed dual problem D_{R} are satisfied by $\boldsymbol{\Omega}$ and all the dual variables are non-negative.

Due to this feasibility, we know that the objective value $\widetilde{\mathsf{D}}_{\mathsf{R}}$ of the relaxed dual problem D_{R}, is no larger than the optimum of D_{R} and thus the optimum $\mathsf{P}_{\mathsf{R}}^{opt}$ of the relaxed primal problem P_{R}. Meanwhile, via the KKT conditions, we can also derive the optimality gap between $\widetilde{\mathsf{D}}_{\mathsf{R}}$ and the objective value $\widetilde{\mathsf{P}}$ of the original problem P achieved by the optimal fractional solution. Combining the above two sides, we can derive the competitive ratio K_1 of the proposed ORFA algorithm in Alg. 1.

To the above end, we first show that the total static cost (i.e., the total cost excludes the switching cost) is upper bounded by the objective value $\widetilde{\mathsf{D}}_{\mathsf{R}}$ of the relaxed dual problem D_{R} achieved by the constructed feasible solution $\boldsymbol{\Omega}\{\widetilde{\boldsymbol{x}}(t),\widetilde{\boldsymbol{y}}(t),\widetilde{\boldsymbol{z}}(t),\widetilde{\boldsymbol{a}}(t),\widetilde{\boldsymbol{u}}(t)\}$, as shown by the following Lemma 1.

Lemma 2: The sum of instance running cost, traffic routing cost and cloud outsourcing cost achieved by ORFA is no larger than the objective value of the relaxed dual problem D_{R} *achieved by the constructed feasible solution* $\boldsymbol{\Omega}$*, i.e.,*

$$\sum_{t\in\mathcal{T}}\Big(C_{I}(t)+C_{O}(t)+C_{R}(t)\Big)\leq\widetilde{\mathsf{D}}_{\mathsf{R}}$$

The instance switching cost is also bounded, as shown by the following Lemma 3.

Lemma 3: The total instance switching cost $\sum_{t\in\mathcal{T}}C_{S}(t)$ *achieved by ORFA is no larger than* $\Big[\ln\Big(1+\frac{C_{\max}}{\epsilon}\Big)+\frac{C_{\max}}{\delta}\Big]$ *times of* D_{R}*, where* $\delta=\min_{t,i,m}\{\widetilde{x}_{i}^{m}(t),\widetilde{x}_{i}^{m}(t)>0\}$ *is the minimum of instances provisioned for any SF, in any edge cloud and at any time slot.*

$$\sum_{t\in\mathcal{T}}C_{S}(t)\leq\Big[\ln\Big(1+\frac{C_{\max}}{\epsilon}\Big)+\frac{C_{\max}}{\delta}\Big]\widetilde{\mathsf{D}}_{\mathsf{R}}$$

The detailed proofs of Lemma 2 and Lemma 3 are given in the Appendix. C and Appendix. D of our online technical report [28], respectively. Based on Lemma 2 and Lemma 3, the overall competitive ratio of the proposed ORFA algorithm is given in the following Theorem 1.

Theorem 1: For the proposed ORFA algorithm, it achieves a competitive ratio of K_1*. That is, the objective value of problem* P *achieved by the optimal fractional solution, denoted by* $\widetilde{\mathsf{P}}$*, is no larger than* K_1 *times of the offline optimum* P^{opt}*, where* K_1 *is given by:*

$$K_{1}=\ln\Big(1+\frac{C_{\max}}{\epsilon}\Big)+\frac{C_{\max}}{\delta}+1$$

Theorem 1 follows immediately Lemma 2 and Lemma 3, since the objective of problem P is the sum of the instances switching cost, instances running cost, traffic routing cost and cloud outsourcing cost.

C. Rounding Gap of RDIP

We next study the rounding gap incurred by the randomized dependent rounding scheme, in terms of competitive ratio K_2 of the cost $\bar{\mathsf{P}}$ achieved by the final rounded solution to the cost $\widetilde{\mathsf{P}}$ achieved by the fractional solution. The basic idea is to leverage the relationship between $\widetilde{\boldsymbol{x}}(t)$ and $\bar{\boldsymbol{x}}(t)$, which has been characterized by the three desirable properties in Sec. IV-B, to establish the connection between their instance running costs $\sum_{i}\sum_{m}b_{i}^{m}(t)\widetilde{x}_{i}^{m}(t)$ and $\sum_{i}\sum_{m}b_{i}^{m}(t)\bar{x}_{i}^{m}(t)$. Then, we further take this connection as a bridge to bound the cost terms in the objective function of the original problem P.

The connection between $\sum_{i}\sum_{m}b_{i}^{m}(t)\widetilde{x}_{i}^{m}(t)$ **and** $\sum_{i}\sum_{m}b_{i}^{m}(t)\bar{x}_{i}^{m}(t)$ We establish this connection based on the weight conservation property discussed in Sec. IV-B. Specifically, recall that after the main loop of each iteration, for each $m\in\mathcal{M}$, the total resource capacity of the two selected elements $i_1,i_2\in\mathcal{I}_{mt}^{-}$ keeps unchange, i.e., $\widetilde{x}_{i_1}^{m}(t)b_{i_1}^{m}+\widetilde{x}_{i_2}^{m}(t)b_{i_2}^{m}=\bar{x}_{i_1}^{m}(t)b_{i_1}^{m}+\bar{x}_{i_2}^{m}(t)b_{i_2}^{m}$. From this equation, we know that after the execution of the RDIP algorithm, for the integral elements $i\in\mathcal{I}_{mt}^{+}$, we have the deterministic equation

$$\sum_{i\in\mathcal{I}_{mt}^{+}}\bar{x}_{i}^{m}(t)b_{i}^{m}=\sum_{i\in\mathcal{I}_{mt}^{+}}\widetilde{x}_{i}^{m}(t)b_{i}^{m}.$$

We also note that after the execution of the RDIP algorithm, there is at most one element remaining in the fractional

[3] For ease of presentation here, we do not write the optimal dual solution in the notation $\boldsymbol{\Omega}\{\cdot\}$

set $\mathcal{I}_{mt}^-$, i.e., $|\mathcal{I}_{mt}^-| \leq 1$. Specifically, if $|\mathcal{I}_{mt}^-| = 1$, for the only element in $\mathcal{I}_{mt}^-$, by defining $\kappa_m = \frac{\max_{i\in\mathcal{I}} C_i^m b_i^m}{\sum_{s\in\mathcal{I}} \beta_s^m [A_s(t)-\widetilde{u}_s(t)]}$, we have

$$\begin{aligned}\sum_{i\in\mathcal{I}_{mt}^-} \bar{x}_i^m b_i^m \leq& \max_{i\in\mathcal{I}} C_i^m b_i^m = \kappa_m \sum_{s\in\mathcal{I}} \beta_s^m [A_s(t) - \widetilde{u}_s(t)] \\ \leq& \kappa_m \sum_{s\in\mathcal{I}} \beta_s^m \widetilde{a}_s(t) \leq \kappa_m \sum_{s\in\mathcal{I}} \sum_{i\in\mathcal{I}} \widetilde{z}_{si}^m(t) \\ \leq& \kappa_m \sum_{i\in\mathcal{I}} \widetilde{x}_i^m(t) b_i^m.\end{aligned}$$

Note that when there is no element in $\mathcal{I}_{mt}^-$, i.e., $|\mathcal{I}_{mt}^-| = 0$, the above inequality still holds since $\sum_{i\in\mathcal{I}_{mt}^-} \bar{x}_i^m(t) b_i^m = 0$.

Combining the above equation and inequality, we have

$$\begin{aligned}\sum_{i\in\mathcal{I}} \bar{x}_i^m(t) b_i^m \leq& \sum_{i\in\mathcal{I}_{mt}^+} \bar{x}_i^m(t) b_i^m + \sum_{i\in\mathcal{I}_{mt}^-} \bar{x}_i^m(t) b_i^m \\ \leq& \sum_{i\in\mathcal{I}_{mt}^+} \widetilde{x}_i^m(t) b_i^m + \kappa_m \sum_{i\in\mathcal{I}} \widetilde{x}_i^m(t) b_i^m \\ \leq& \sum_{i\in\mathcal{I}_{mt}} \widetilde{x}_i^m(t) b_i^m + \kappa_m \sum_{i\in\mathcal{I}} \widetilde{x}_i^m(t) b_i^m \\ =& (1+\kappa_m) \sum_{i\in\mathcal{I}} \widetilde{x}_i^m(t) b_i^m.\end{aligned}$$

By summing over the above inequality over all the SFs $m \in \mathcal{M}$ and defining $\kappa = \max_{m\in\mathcal{M}} \kappa_m$, we further have

$$\begin{aligned}\sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} \bar{x}_i^m(t) b_i^m \leq& \sum_{m\in\mathcal{M}} (1+\kappa_m) \sum_{i\in\mathcal{I}} \widetilde{x}_i^m(t) b_i^m \\ \leq& (1+\kappa) \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} \widetilde{x}_i^m(t) b_i^m. \qquad (8)\end{aligned}$$

1) Bounding the Instance Running Cost: Based on the inequality Eq. (8), we bound the instance running cost $\sum_t \sum_i \sum_m p_i^m(t) \bar{x}_i^m(t)$ at each time slot t as follows:

$$\begin{aligned}&\sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} \bar{x}_i^m(t) p_i^m(t) \\ \leq& \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} \bar{x}_i^m(t) b_i^m \frac{p_i^m(t)}{b_i^m} \\ \leq& \max_{t,i,m} \frac{p_i^m(t)}{b_i^m} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} \bar{x}_i^m(t) b_i^m \\ \leq& \max_{t,i,m} \frac{p_i^m(t)}{b_i^m} (1+\kappa) \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} b_i^m \widetilde{x}_i^m(t) \\ =& \max_{t,i,m} \frac{p_i^m(t)}{b_i^m} (1+\kappa) \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} p_i^m(t) \widetilde{x}_i^m(t) \frac{b_i^m}{p_i^m(t)} \\ \leq& (1+\kappa) \max_{t,i,m} \frac{p_i^m(t)}{b_i^m} \max_{t,i,m} \frac{b_i^m}{p_i^m(t)} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} p_i^m(t) \widetilde{x}_i^m(t).\end{aligned}$$

2) Bounding the Instance Switching Cost: We bound the instance switching cost $\sum_t \sum_i \sum_m q_i^m [\bar{x}_i^m(t) - \bar{x}_i^m(t-1)]^+$ as follows

$$\begin{aligned}&\sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} q_i^m [\bar{x}_i^m(t) - \bar{x}_i^m(t-1)]^+ \\ \leq& \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} q_i^m \bar{x}_i^m(t) = \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} b_i^m \bar{x}_i^m(t) \frac{q_i^m}{b_i^m} \\ \leq& \max_{i,m} \frac{q_i^m}{b_i^m} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} b_i^m \bar{x}_i^m(t) \\ \leq& (1+\kappa) \max_{t,i,m} \frac{b_i^m}{p_i^m(t)} \max_{i,m} \frac{q_i^m}{b_i^m} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} p_i^m(t) \widetilde{x}_i^m(t)\end{aligned}$$

3) Bounding the Traffic Routing Cost: The traffic routing cost can be bounded as follows

$$\begin{aligned}&\sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{I}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} \Big(g_{si}^m \bar{z}_{si}^m(t) + \sum_{n\in\mathcal{M}} \sum_{j\in\mathcal{I}} h_{mn}^s l_{ij} \bar{y}_{sij}^{mn}(t) \Big) \\ \leq& \max_{s,i,m} g_{si}^m \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{I}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} \bar{z}_{si}^m(t) \\ &+ \max_{i,j} l_{ij} \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{I}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} \sum_{n\in\mathcal{M}} \sum_{j\in\mathcal{I}} h_{mn}^s \bar{y}_{sij}^{mn}(t) \\ \leq& \max_{s,i,m} g_{si}^m \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} b_i^m \bar{x}_i^m(t) \\ &+ \max_{i,j} l_{ij} \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{I}} \sum_{n\in\mathcal{M}} \sum_{j\in\mathcal{I}} \bar{z}_{sj}^n(t) \\ =& \max_{s,i,m} g_{si}^m \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} b_i^m \bar{x}_i^m(t) \\ &+ \max_{i,j} l_{ij} \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{I}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} \bar{z}_{si}^m(t) \\ \leq& (\max_{s,i,m} g_{si}^m + \max_{i,j} l_{ij}) \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}} b_i^m \bar{x}_i^m(t) \\ \leq& (1+\kappa) \max_{t,i,m} \frac{b_i^m}{p_i^m(t)} \\ &\times (\max_{s,i,m} g_{si}^m + \max_{i,j} l_{ij}) \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} p_i^m(t) \widetilde{x}_i^m(t)\end{aligned}$$

Finally, for the cloud outsourcing cost, recall that due to the weight conservation property, the total resource capacity of the rounded solution is no smaller than that of the optimal fractional solution. Therefore, when redirecting the traffic after the rounding, we do not outsource additional traffic to the cloud for processing, meaning that the cloud outsourcing cost of the rounded solution is the same with that of the optimal fractional solution.

Based on the above intermediate results that maps the cost terms of the rounded solution to that of the optimal fractional solution, the competitive ratio of the proposed RDIP algorithm is now given in the following Theorem 2.

Theorem 2: For the proposed RDIP algorithm, it achieves a competitive ratio of K_2. That is, the objective value $\bar{\mathsf{P}}$ of problem P achieved by the rounded solution is no larger than K_2 times of $\widetilde{\mathsf{P}}$ achieved by the optimal fractional solution, *where* $K_2 = \xi_1 + \xi_2 + \xi_3$, *and*

$$\xi_1 = (1+\kappa)\max_{t,i,m}\frac{p_i^m(t)}{b_i^m}\max_{t,i,m}\frac{b_i^m}{p_i^m(t)},$$

$$\xi_2 = (1+\kappa)\max_{t,i,m}\frac{b_i^m}{p_i^m(t)}\max_{i,m}\frac{q_i^m}{b_i^m},$$

$$\xi_3 = (1+\kappa)\max_{t,i,m}\frac{b_i^m}{p_i^m(t)}(\max_{s,i,m} g_{si}^m + \max_{ij} l_{ij}).$$

D. The Overall Competitive Ratio

Based on the competitive ratio of the proposed ORFA and RDIP algorithms, given in Theorem 1 and Theorem 2, respectively, we give the overall competitive ratio of the online algorithm for joint instance provisioning and traffic routing in the following Theorem 3.

Theorem 3: For the proposed online algorithm for joint instance provisioning and traffic routing, it achieves a competitive ratio of $K_1\ K_2$. *That is, the objective value* $\bar{\mathsf{P}}$ *of problem* P *achieved by the rounded solution is no larger than* K_1K_2 *times of the offline optimum, where* K_1 *and* K_2 *are derived in Theorem 1 and Theorem 2, respectively.*

Theorem 3 follows immediately the chain of inequalities in Eq. (4). We now discuss some insights on the final competitive ratio K_1K_2.

Firstly, the final competitive ratio decreases with the increasing of the tunable parameter ϵ. By increasing ϵ to be large enough, we can push the competitive ratio K_1 given in Theorem 2 arbitrarily close to $\frac{C_{\max}}{\delta}+1$. However, overly aggressive increases of the control parameter ϵ can also increase the time complexity of the regularized problem P_{Re}, since the convexity of P_{Re} deteriorates as ϵ grows (according to the decreasing of the second-order derivative of the objective function). Therefore, the control parameter ϵ works as a knob to balance the performance-complexity tradeoff.

Secondly, recall that when performing traffic re-routing based on $\bar{\boldsymbol{x}}(t)$ to obtain a complete feasible solution, we exploit the weight conservation property to modify $\widetilde{\boldsymbol{y}}(t)$ and $\widetilde{\boldsymbol{z}}(t)$, while keeping $\widetilde{\boldsymbol{u}}(t)$ and $\widetilde{\boldsymbol{a}}(t)$ unchanged. Compared to the naive approach that directly brings $\bar{\boldsymbol{x}}(t)$ back to the original problem P to obtain the complete feasible solution, the above applied approach has lower time complexity but also reduced optimality. Therefore, the derived ratio K_1K_2 can also serve as an upper bound of the competitive ratio of the scheme that applies the above naive approach to traffic re-routing.

Finally, interestingly, we observe that the derived competitive ratio is deterministic, though the proposed rounding scheme is random in nature. The rationale is that it is derived based on the weight conservation property, which is actually deterministic rather than random. Furthermore, if we leverage the marginal distributed property which ensures $\mathbb{E}\{\bar{x}_i^m(t)\} = \widetilde{x}_i^m(t)$ after the main loop of the RDIP algorithm, we may derive a random competitive ratio. We leave this horizontal route as an extension to be addressed in the further work.

VI. Performance Evaluation

In this section, we conduct trace-driven simulations to evaluate the practical benefits of the proposed online orchestration framework. The simulations are based on real-world workload traces and electricity prices.

TABLE I

Configuration of 4 Required SFs

SF	Change ratio α	Instance capacity b
Firewall	0.8-1.0	30
Proxy	1.0	30
NAT	0.6	20
IDS	0.8-1.0	20

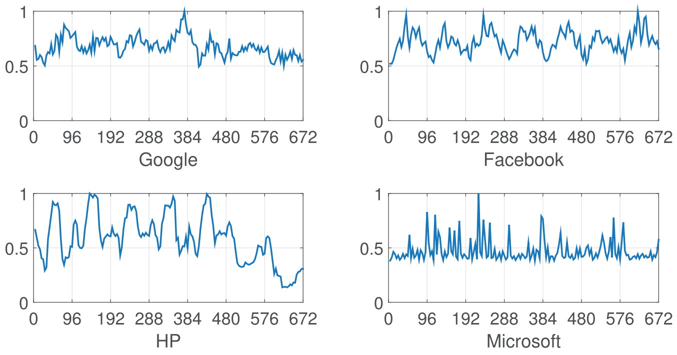

Fig. 4. The workload trace of Google, Facebook, HP and Microsoft datacenters.

A. Experimental Setup

1) Infrastructure: We simulate a cross-edge system that deploys $M = 4$ edge clouds in the geographical center of four regions in New York city: Manhattan, Brooklyn, Queues and Bronx. The input traffic at each edge cloud requires a SFC that contains 3 SFs randomly chosen from the 4 extensively studied virtual network functions (VNFs) in Table I.

2) Workload Trace: Since edge computing is still in a very early stage, there is no public accessible workload trace from an edge cloud. In response, we adopt the workload trace of mega-scale datacenters to simulate the traffic arrival at each edge cloud. Specifically, we associate to the one-week hourly traces of the datacenters of Google [32], Facebook [33], HP [34] and Microsoft [35] to each of the 4 edge cloud, respectively. The normalized traffic arrival of the traces is shown in Fig. 4. In the following simulations, we proportionally scale the normalized trace up to 1000 times to represent the actual traffic arrival at each edge cloud.

3) Real Cost Data: To incorporate the spatial and temporal diversities of the instance running cost $p_i^m(t)$, here we use the product of the instance capacity and the regional electricity price to simulate the instance running cost $p_i^m(t)$. Specifically, We download the hourly locational marginal prices (LMP) in real-time electricity markets of the 4 regions from the website of NYISO (New York Independent System Operator) [36]. The time period of this data is October 10-16, 2018, including one week or 168 hours.

4) System Parameters: We set C_i^m, the maximal number of available instances of SF m at edge cloud i, to be the minimal integer that can serve 80% of the peak-arrival at each edge cloud locally. For the cross-edge traffic routing cost parameter d_{ij}, we assume it is proportional to the geographical distance, and the total traffic routing cost $C_R(t)$ has a similar

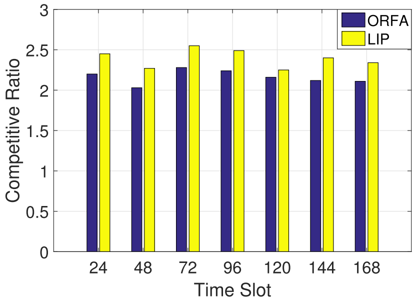

Fig. 5. The competitive ratio of different online algorithms.

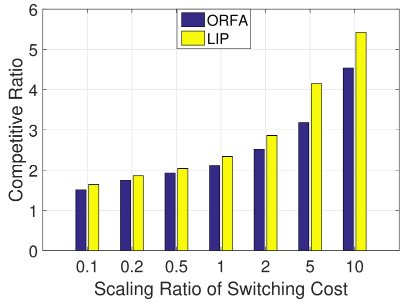

Fig. 6. The effect of the switching cost on the competitive ratio.

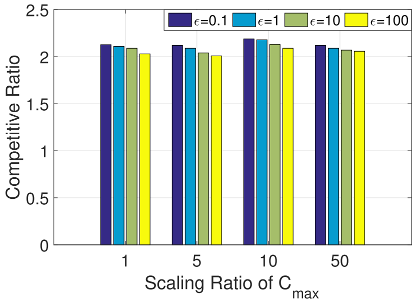

Fig. 7. The effect of the control parameter ϵ on the competitive ratio.

order of magnitude to the instance running cost $C_I(t)$. For the switching cost parameter q_i^m, we also set it to make the switching cost $C_S(t)$ has a similar order of magnitude to $C_I(t)$. For the cloud outsourcing cost parameter r_s, we set it is sufficient large, such that the central cloud would only be used in the presence of flash crowd.

5) Benchmarks: To empirically evaluate the competitive ratio of the proposed online framework, we adopt the state-of-the-art MILP solver Gurobi to obtain the offline optimum of the long-term cost minimization problem P. To demonstrate the efficacy of the online algorithm ORFA, we compare it to the lazy instance provisioning (LIP) that has been extensively applied to decouple time-coupling term in literature [37], [38]. Furthermore, to demonstrate the efficacy of the rounding scheme RDIP, we compare it to another three benchmarks: (1) the randomized independent instance provisioning (RIIP) [31] as we have discussed in the first paragraph of Sec. IV-B. (2) EC-Greedy approach which directly rounds-up all the fractional solutions to process the absorbed traffic by the edge clouds. (3) CC-Greedy approach which directly rounds-down all the fractional solutions, and using the central cloud to serve the traffic which can not be covered by the edge clouds.

B. Evaluation Results

1) Efficiency of the Online Algorithm: We first examine the competitive ratio achieved by the online algorithm ORFA (e.g., Alg. 1) without rounding the factional solution, which is the ratio of the total cost achieved by ORFA to that achieved by the offline minimum cost. For comparison, we plot the competitive ratio of ORFA as well as the benchmark LIP in Fig. 5, under varying number of total time slots. From this figure, we observe that: (1) ORFA always outperforms LIP that has been widely applied in literature [37], [38], demonstrating the effectiveness of our proposed regularization-based online algorithm. (2) As the number of total time slots varies, the competitive ratio of ORFA only changes very slightly, indicating that ORFA has stable performance against varying time span.

2) Effect of the Switching Cost: We continue to investigate the effect of the instance switching cost (i.e., q_i^m in our formulation) on the competitive ratio of ORFA and LIP, by multiplying the switching cost of each SF instance with a various scaling ratios. The results are plotted in Fig. 6, which shows that as the switching cost increases, the competitive ratios of both ORFA and LIP increase dramatically. The rationale is that, by increasing the switching cost q_i^m, we would increasingly focus on minimizing the total switching cost. However, since the switching cost term is time-coupling and involves future stochastic information, the optimality gap incurred by any online algorithm would increase as the switching cost q_i^m grows. Since the weights of other cost terms keep unchanged, the optimality gap of the switching cost term would deteriorate the overall competitive ratio of the online algorithm.

3) Sensibility to Varying Parameters: Recall that in Theorem 1, the worst-case competitive ratio of ORFA is parameterized by the control parameter ϵ and $C_{\max}$, the maximum number of available instance for each SF m at each edge cloud i. We now examine the effect of parameters ϵ and $C_{\max}$ on the actual competitive ratio of ORFA. Fig. 7 depicts the actual competitive ratio under various ϵ and $C_{\max}$, from which we observe that: (1) as the control parameter ϵ increases, the competitive ratio descents. This quantitatively corroborates the insights we unfolded in Sec. V-D. However, we should also note that while ϵ varies significantly, the competitive ratio only changes slightly, meaning that the effect of ϵ on the competitive ratio is very limited. (2) As $C_{\max}$ changes, the competitive ratio remains relatively stable. This suggest that in practice, the impact of $C_{\max}$ is not obvious.

4) Competitive Ratio of the Rounding Scheme: After executing the online algorithm ORFA to obtain the fractional solution, we need to round it to integer solution for instance provisioning. Here we compare the competitive ratio of different rounding schemes in Fig. 8, it demonstrates that the competitive ratio of our proposed RDIP is relatively stable and substantially outperforms those of the three benchmarks. Here we also note that the independent rounding scheme RDIP outperforms the other two greedy strategies, due to the fact that

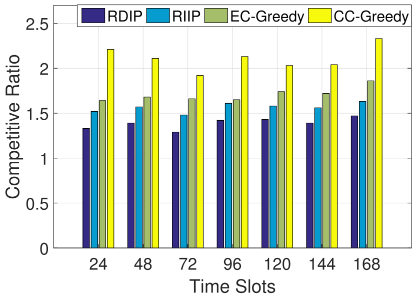

Fig. 8. The competitive ratio of various rounding schemes.

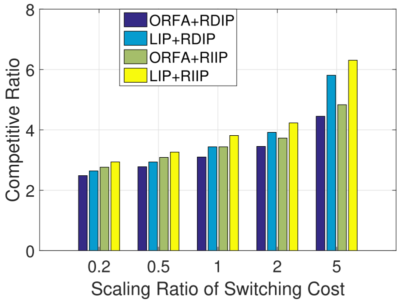

Fig. 9. The overall competitive ratio of different algorithm combinations.

the optimality introduced by the rounding can be compensated by the traffic-rerouting conducted afterwards.

5) Examining the Overall Competitive Ratio: We finally verify the performance of our complete online algorithm for joint instance provisioning and traffic routing, by comparing the overall competitive ratio of different combinations of the online algorithm and the rounding scheme. The results in Fig. 9 show that the combination of our proposed online algorithm and rounding scheme performs the best. While in a sharp contrast, the combination of the benchmarks leads to the worst performance. This confirms the efficacy of both the proposed ORFA algorithm and the RDIP rounding scheme. We also observe that similar to Fig. 6, the competitive ratio ascents as the switching cost q_i^m increases, due to the aforementioned fact that the optimality gap of the switching cost term rises as the switching cost q_i^m increases.

VII. Concluding Remarks

This work presented an online orchestration framework for cross-edge service function chaining. It jointly optimizes the resource provisioning and traffic routing to maximize the holistic cost-efficiency. Since the formulated optimization problem is NP-hard and involves future uncertain information, a joint optimization framework is designed, which carefully blends the advantages of an online optimization technique and an approximate optimization method. Specifically, by adopting a regularization technique, we decompose the long-term problem into a series of one-shot fractional problems which can be readily solved. With a randomized dependent scheme, we further round the fractional solutions to a near-optimal integral solution of the original problem. The resulting joint online algorithm achieves a good performance guarantee, as verified by both theoretical analysis and trace-driven simulations.

References

[1] 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, 3rd Quart., 2017.

[2] G. Ananthanarayanan *et al.*, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, Oct. 2017.

[3] N. Chen, Y. Yang, T. Zhang, M.-T. Zhou, X. Luo, and J. K. Zao, "Fog as a service technology," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 95–101, Nov. 2018.

[4] 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*, to be published.

[5] X. Chen, Z. Zhou, W. Wu, D. Wu, and J. Zhang, "Socially-motivated cooperative mobile edge computing," *IEEE Netw.*, no. 32, no. 6, pp. 177–183, Nov./Dec. 2018.

[6] L. Jiao, L. Pu, L. Wang, X. Lin, and J. Li, "Multiple granularity online control of cloudlet networks for edge computing," in *Proc. IEEE SECON*, Jun. 2018, pp. 1–9.

[7] *ML on the Edge*. Accessed: Nov. 28, 2018. [Online]. Available: https://drive.google.com/file/d/0B2A84I7Zi4zTGM4cU5qNVF2OUE/view

[8] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.

[9] L. Jiao, A. M. Tulino, J. Llorca, Y. Jin, and A. Sala, "Smoothed online resource allocation in multi-tier distributed cloud networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2556–2570, Aug. 2017.

[10] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *J. ACM*, vol. 53, no. 3, pp. 324–360, 2006.

[11] H. Hantouti, N. Benamar, T. Taleb, and A. Laghrissi, "Traffic steering for service function chaining," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 487–507, 1st Quart., 2018.

[12] X. Fei, F. Liu, H. Xu, and H. Jin, "Towards load-balanced vnf assignment in geo-distributed nfv infrastructure," in *Proc. IEEE/ACM IWQoS*, Jun. 2017, pp. 1–10.

[13] R. Zhou, "An online placement scheme for VNF chains in geo-distributed clouds," in *Proc. IEEE IWQoS*, Jun. 2018, pp. 1–2.

[14] M. Abu-Lebdeh, D. Naboulsi, R. H. Glitho, and C. W. Tchouati, "On the placement of VNF managers in large-scale and distributed NFV systems," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 875–889, Dec. 2017.

[15] I. Benkacem, T. Taleb, M. Bagaa, and H. Flinck, "Optimal VNFs placement in CDN slicing over multi-cloud environment," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 616–627, Mar. 2018.

[16] Y. Jia, C. Wu, Z. Li, F. Le, and A. Liu, "Online scaling of NFV service chains across geo-distributed datacenters," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 699–710, Apr. 2018.

[17] L. Dinh-Xuan, M. Seufert, F. Wamser, P. Tran-Gia, C. Vassilakis, and A. Zafeiropoulos, "Performance evaluation of service functions chain placement algorithms in edge cloud," in *Proc. 30th Int. Teletraffic Congr. (ITC)*, Sep. 2018, pp. 227–235.

[18] R. Gouareb, V. Friderikos, and A.-H. Aghvami, "Virtual network functions routing and placement for edge cloud latency minimization," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2346–2357, Oct. 2018.

[19] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, "Towards edge slicing: VNF placement algorithms for a dynamic & realistic edge cloud environment," in *Proc. IEEE GLOBECOM*, Dec. 2017, pp. 1–6.

[20] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware VNF placement for service-customized 5G network slices," in *Proc. IEEE INFOCOM*, Apr./May 2019, pp. 2449–2457.

[21] A. Gupta, B. Jaumard, M. Tornatore, and B. Mukherjee, "A scalable approach for service Chain mapping with multiple SC instances in a wide-area network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 529–541, Mar. 2018.

[22] L. Qu, M. Khabbaz, and C. Assi, "Reliability-aware service chaining in carrier-grade softwarized networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 558–573, Mar. 2018.

[23] C. Mouradian, S. Kianpisheh, M. Abu-Lebdeh, F. Ebrahimnezhad, N. T. Jahromi, and R. H. Glitho, "Application component placement in NFV-based hybrid cloud/fog systems with mobile fog nodes," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1130–1143, May 2019.

[24] T. Taleb, P. A. Frangoudis, I. Benkacem, and A. Ksentini, "CDN slicing over a multi-domain edge cloud," *IEEE Trans. Mobile Comput.*, to be published.

[25] X. Chen, W. Li, S. Lu, Z. Zhou, and X. Fu, "Efficient resource allocation for on-demand mobile-edge cloud computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8769–8780, Sep. 2018.

[26] Z. Zhou, F. Liu, S. Chen, and Z. Li, "A truthful and efficient incentive mechanism for demand response in green datacenters," *IEEE Trans. Parallel Distrib. Syst.*, to be published.

[27] V. V. Vazirani, *Approximation Algorithms*. Berlin, Germany: Springer, 2013.

[28] *Technical Report for Online Orchestration of Cross-Edge Service Function Chaining for Cost-Efficient Edge Computing*. Accessed: Jun. 2019. [Online]. Available: https://1drv.ms/b/s!Ar9mS_s-frkZgctM5aAhsakHzqE-bw

[29] N. Buchbinder, S. Chen, and J. S. Naor, "Competitive analysis via regularization," in *Proc. ACM/SIAM SODA*, 2014, pp. 436–444.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[31] P. Raghavan and C. D. Tompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.

[32] *Google Cluster Data*. Accessed: Jan. 2010. [Online]. Available: https://code.google.com/p/googleclusterdata/

[33] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating mapreduce performance using workload suites," in *Proc. IEEE MASCOTS*, Jul. 2011, pp. 390–399.

[34] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew, "Geographical load balancing with renewables," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 39, no. 3, pp. 62–66, 2011.

[35] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: Practical power management for enterprise storage," in *Proc. USENIX Conf. File Storage Technol. (FAST)*, 2008, pp. 1–15.

[36] NYISO. *Energy Market & Operational Data*. Accessed: May 2018. [Online]. Available: https://www.nyiso.com/energy-market-operational-data

[37] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1098–1106.

[38] L. Zhang *et al.*, "Moving big data to the cloud: An online cost-minimizing approach," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2710–2721, Dec. 2013.

Zhi Zhou received the B.S., M.E., and Ph.D. degrees from the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), Wuhan, China, in 2012, 2014 and 2017, respectively. In 2016, he has been a Visiting Scholar with the University of Gottingen. He is currently a Research Fellow with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His research interests include edge computing, cloud computing, and distributed systems. He was a recipient of the 2018 ACM Wuhan and Hubei Computer Society Doctoral Dissertation Award and the Best Paper Award from the IEEE UIC 2018. He was a General Co-Chair of the 2018 International Workshop on Intelligent Cloud Computing and Networking (ICCN).

Qiong Wu received the B.S. degree from the School of Data and Computer Science, Sun Yat-sen University (SYSU), Guangzhou, China, in 2017, where she is currently pursuing the M.S. degree with the School of Data and Computer Science. Her primary research interests include social data analysis, data-driven modeling, and mobile edge computing.

Xu Chen received the Ph.D. degree in information engineering from The Chinese University of Hong Kong in 2012. He was a Post-Doctoral Research Associate with Arizona State University, Tempe, AZ, USA, from 2012 to 2014, and a Humboldt Scholar Fellow with the Institute of Computer Science, University of Goettingen, Germany, from 2014 to 2016. He is currently a Full Professor with Sun Yat-sen University, Guangzhou, China, and the Vice Director of the National and Local Joint Engineering Laboratory of Digital Home Interactive Applications. He received the prestigious Humboldt Research Fellowship from the Alexander von Humboldt Foundation of Germany, the 2014 Hong Kong Young Scientist Runner-up Award, the 2016 Thousand Talents Plan Award for Young Professionals of China, the 2017 IEEE Communication Society Asia-Pacific Outstanding Young Researcher Award, the 2017 IEEE ComSoc Young Professional Best Paper Award, the Honorable Mention Award from the 2010 IEEE International Conference on Intelligence and Security Informatics (ISI), the Best Paper Runner-up Award from the 2014 IEEE International Conference on Computer Communications (INFOCOM), and the Best Paper Award from the 2017 IEEE International Conference on Communications (ICC). He is currently an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Series on Network Softwarization and Enablers.